Przetwarzanie dźwięków i obrazów

Filtry cyfrowe

część 4:

SPECJALNE TYPY FILTRÓW

i ich zastosowania w przetwarzaniu dźwięków

Opracowanie: Grzegorz Szwoch

Politechnika Gdańska, Katedra Systemów Multimedialnych

greg@multimed.org

- W poprzednich wykładach przedstawiliśmy typowe filtry FIR i IIR tłumiące zakresy częstotliwości, a także filtry adaptacyjne.
- Na tym wykładzie przedstawione będą specyficzne typy filtrów do konkretnych zastosowań, związanych z przetwarzaniem dźwięku, w tym:
  - filtry uśredniające,
  - filtry różniczkujące, całkujące i Hilberta,
  - filtry grzebieniowe i wszechprzepustowe,
  - filtry parametryczne (półkowe, szczytowe),
  - filtry interpolacyjne, decymacyjne i polifazowe.

Przypomnijmy filtr uśredniający metodą średniej ruchomej (MA):

$$y[n]=h\cdot x[n]+h\cdot x[n-1]+h\cdot x[n-2]+\ldots+h\cdot x[n-(N-1)]$$

gdzie $h = 1 / N$.

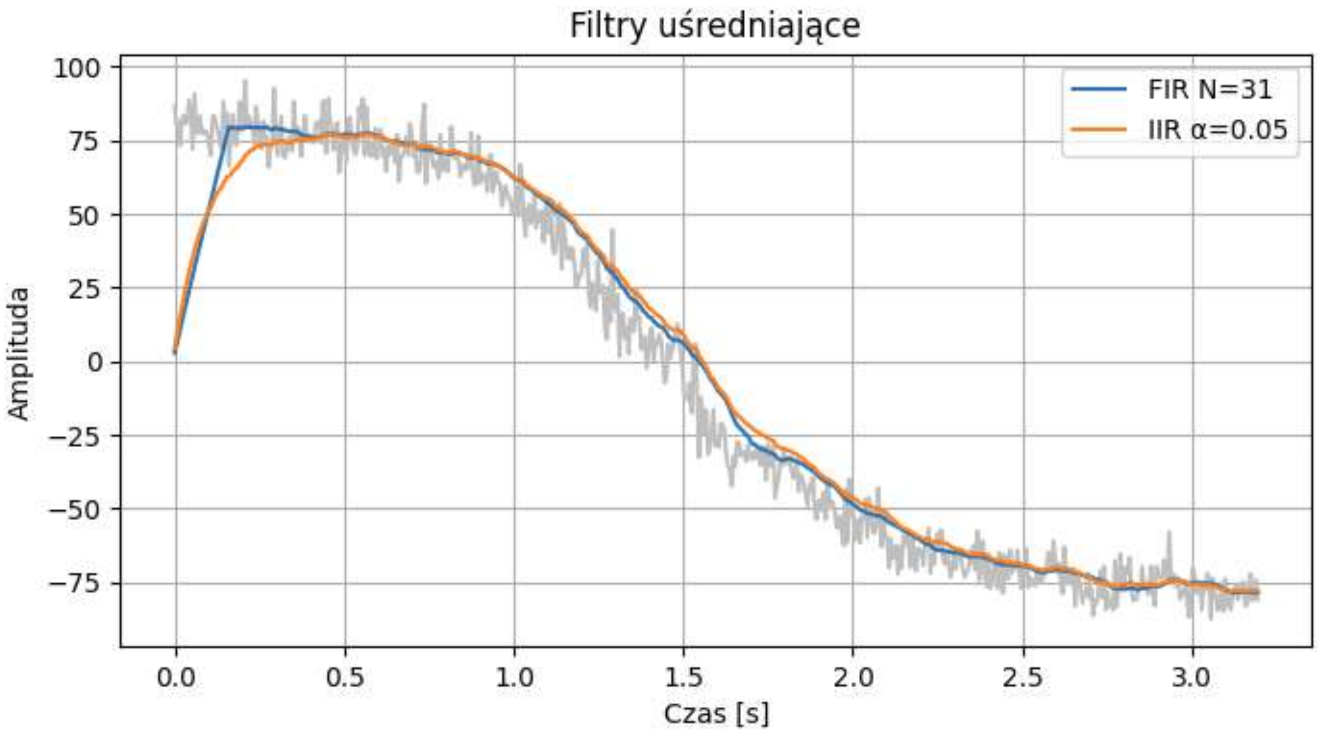
Jest to filtr typu FIR. Wada: konieczność pamiętania N-1 poprzednich próbek i używania ich do obliczeń.

Filtr uśredniający zrealizowany jako filtr IIR:

$$y[n] = \alpha\, x[n] + (1-\alpha)\, y[n-1]$$

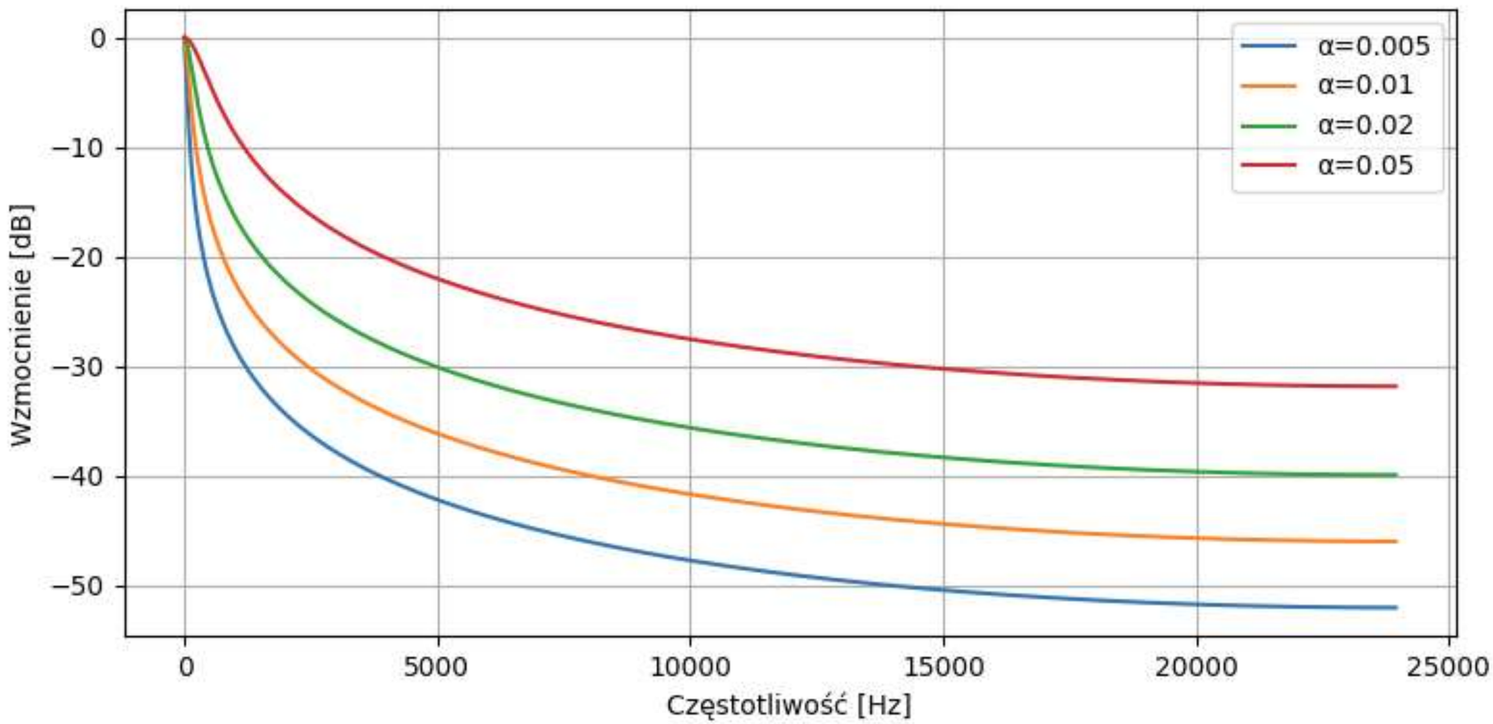
Transmitancja filtru:

$$H(z) = \frac{\alpha}{1-(1-\alpha)z^{-1}}$$

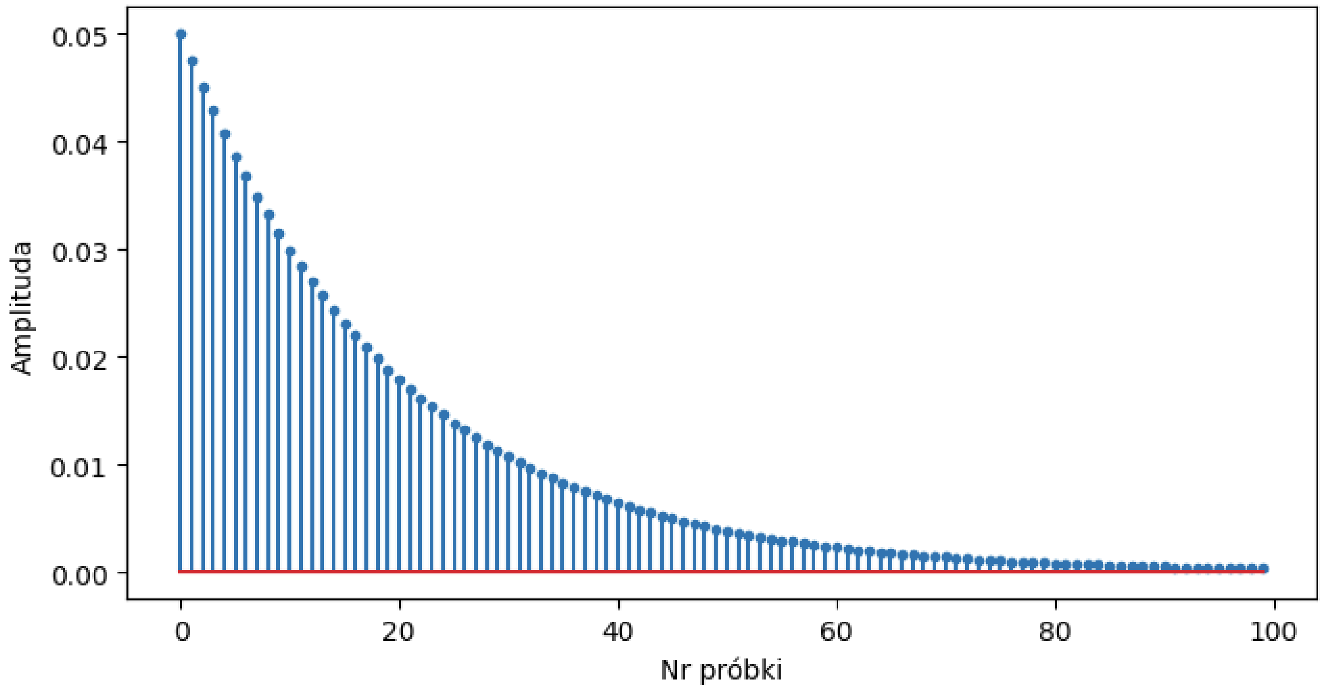


Filtr wymaga pamiętania tylko poprzedniej wartości wyjściowej *y*[*n*-1] i wykonania jednego mnożenia.

Charakterystyki częstotliwościowe filtru
- zależą od współczynnika α – mała liczba dodatnia.



Odpowiedź impulsowa – maleje wykładniczo:



Filtr uśredniający omawianego typu nazywa się filtrem uśredniania wykładniczego (*exponential averaging filter*).

Jeżeli wartość wejściowa (*x*) zmieni się o 1,
to czas potrzebny na to, aby wartość wyjściowa (*y*) zmieniła się o:

$$\Delta y = 1 - \frac{1}{e} \cong 0{,}632$$

nazywa się stałą czasową τ.

Po upływie czasu 5τ, *y* osiąga około 0,9933.

Wartość α potrzebna do uzyskania żądanej stałej czasowej τ jest równa:

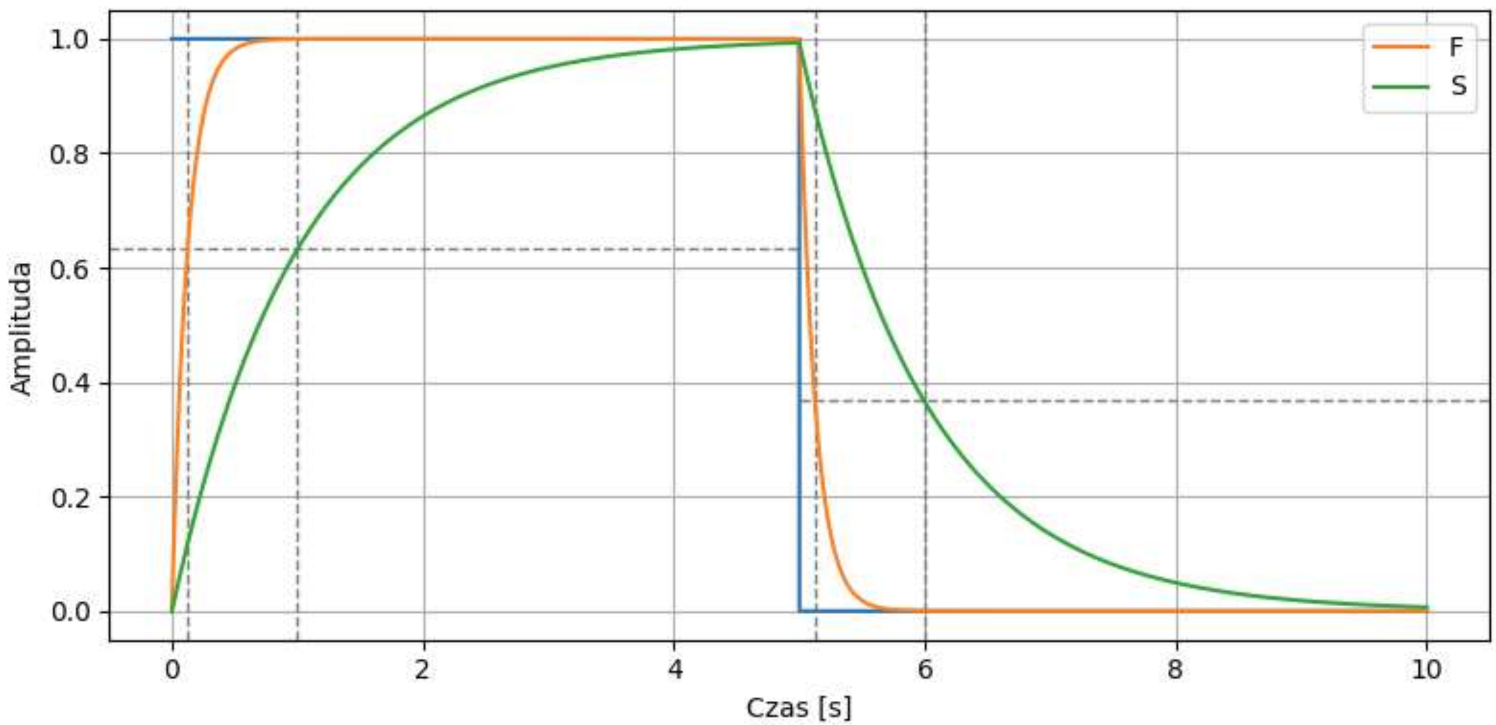
$$\alpha = e^{-\frac{T}{\tau}}$$

gdzie *T* jest odstępem między próbkami (*T* = 1 / *fs*).

Praktyczny przykład: miernik poziomu dźwięku.

Norma definiuje stałe czasowe uśredniania:

- *Fast* (F): τ = 0,125 s = 1/8 s
- *Slow* (S) : τ = 1 s



Filtr różniczkujący (*differentiator*)

- Oblicza pochodną sygnału.
- W sygnale cyfrowym pochodna oznacza po prostu różnicę między wartościami kolejnych próbek.
- Najprostszy, „naiwny” układ różniczkujący to filtr FIR pierwszego rzędu, o równaniu:

$$y[n] = x[n] - x[n-1]$$

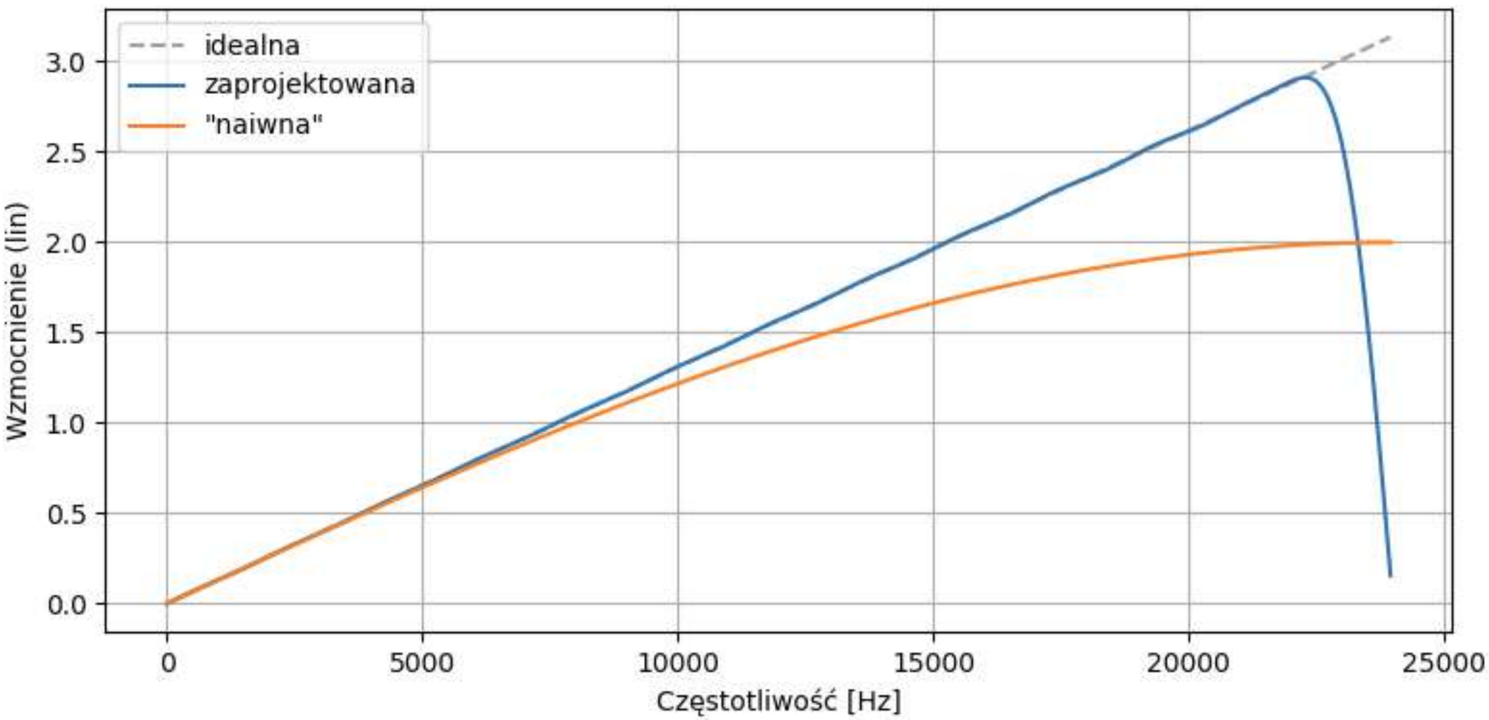
- Można też tworzyć filtry wykorzystujące więcej niż jedna próbkę do obliczeń, z wagami.

- Idealny filtr różniczkujący posiada transmitancję:

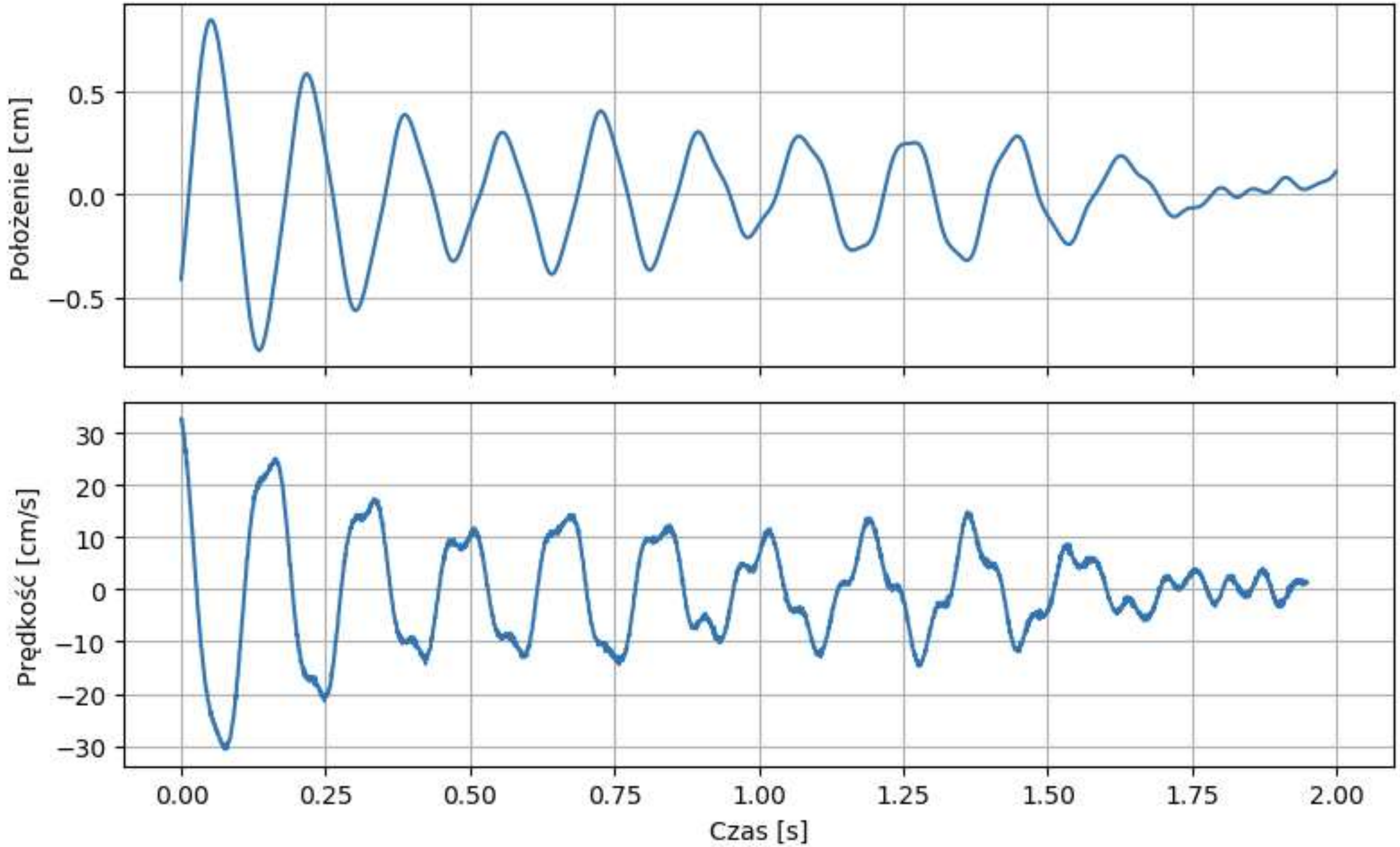
$$H(z) = j\omega$$

- Widmo amplitudowe narasta liniowo od (0, 0) do (π, 2π).
- Można zaprojektować taki filtr FIR, podając punkty charakterystyki, np.: (0, 0); (0.99999, 2*pi), (1, 0).
- Odpowiedź impulsową przycinamy do żądanej długości.
- Odpowiedź impulsowa musi być antysymetryczna!
- Zatem musi to być filtr typu III (nieparzysty), ew. typu IV (parzysty).

Charakterystyka widmowa filtru różniczkującego - mniejsza długość filtru to większe odchylenie na końcu pasma.



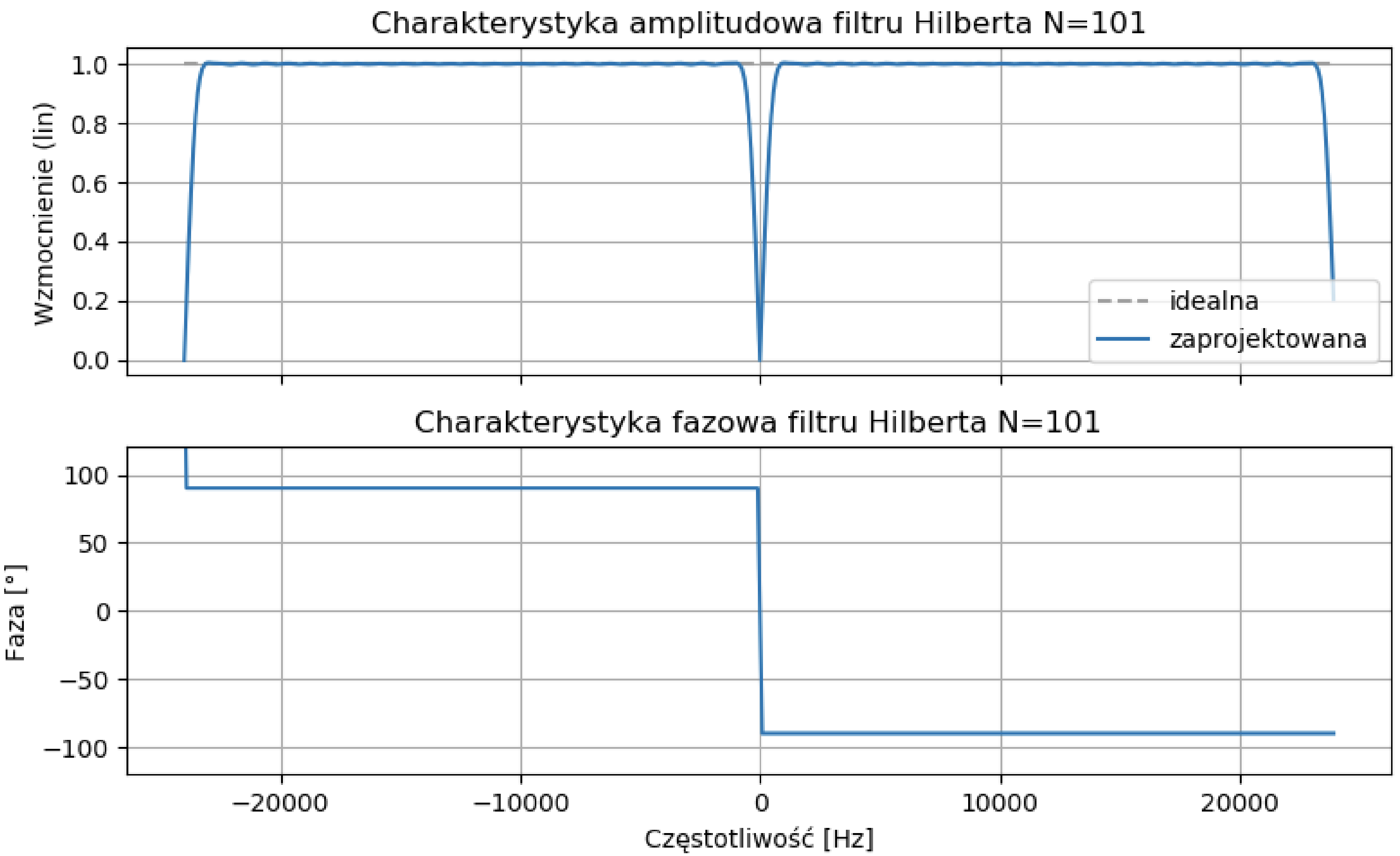
Przykład zastosowania: filtrujemy sygnał położenia obiektu. Pochodna położenia = prędkość.



Filtr Hilberta – wykonuje przesunięcie fazy o 90°.

- Transmitancja: $-j$ dla dodatnich częstotliwości, j dla ujemnych, 0 dla zera.
- Moduł transmitancji: 1 w całym paśmie (filtr nie modyfikuje amplitud widma).
- Możemy zaprojektować filtr tak, że ma zerowe wzmocnienie dla 0 i π, wzmocnienie 1 pomiędzy: (0, 0); (0.00001, 1); (0.99999, 1); (1, 0).
- Również tutaj odpowiedź impulsowa musi być antysymetryczna (typ III lub IV).

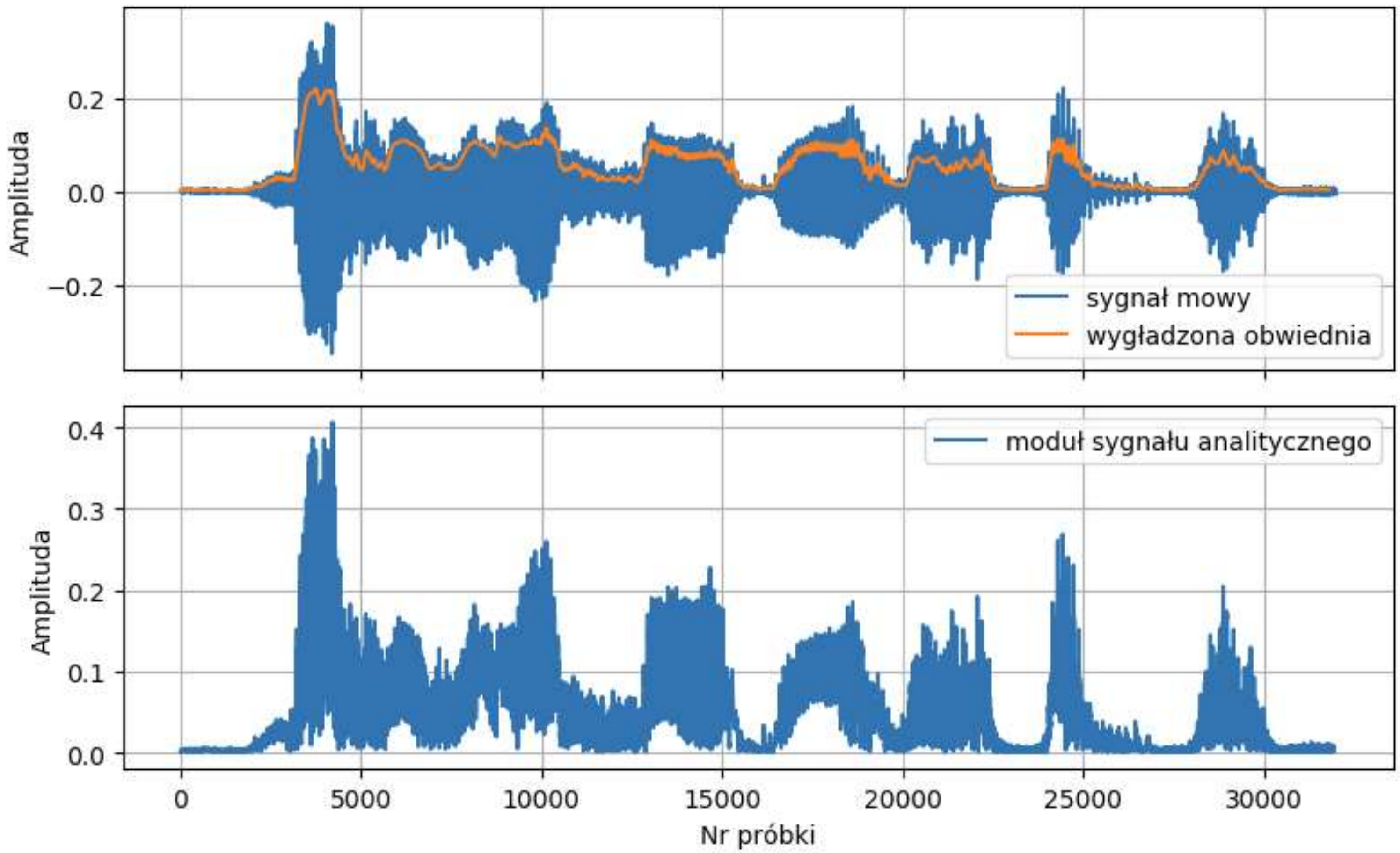
Filtr Hilberta – wykres charakterystyki amplitudowej i skorygowanej charakterystyki fazowej



Po co stosować filtr Hilberta?

- Sygnał analityczny to sygnał zespolony, w którym:
  - część rzeczywista = oryginalny sygnał,
  - część urojona = sygnał po filtracji Hilberta.
- Dla sygnałów w przybliżeniu symetrycznych, np. dla dźwięku:
  - moduł s.a. = obwiednia, czyli reprezentacja zmian energii sygnału (np. głośności dźwięku),
  - faza s.a. = chwilowa częstotliwość.
- Sygnały analityczne używane są w analizie sygnałów oraz do demodulacji sygnałów.

Przykład praktyczny: moduł sygnału analitycznego daje obwiednię sygnału mowy. Po jej wygładzeniu za pomocą np. filtru DP dostajemy sygnał reprezentujący głośność.



Filtr całkujący (*integrator*)

- Przeciwieństwo filtru różniczkującego.
- W sygnale cyfrowym całkowanie oznacza po prostu sumowanie wartości kolejnych próbek.
- Idealny filtr całkujący ma równanie:

$$y[n] = x[n] + y[n-1]$$

- Jest to układ rekursywny (IIR), który sumuje próbki sygnału z wejścia.
- Nie da się takiego filtru zrealizować w praktyce, ponieważ nie jest stabilny!

- Rozwiązanie problemu stabilności jest proste: pozwalamy, aby filtr „zapomniał” część obliczonej poprzednio sumy:

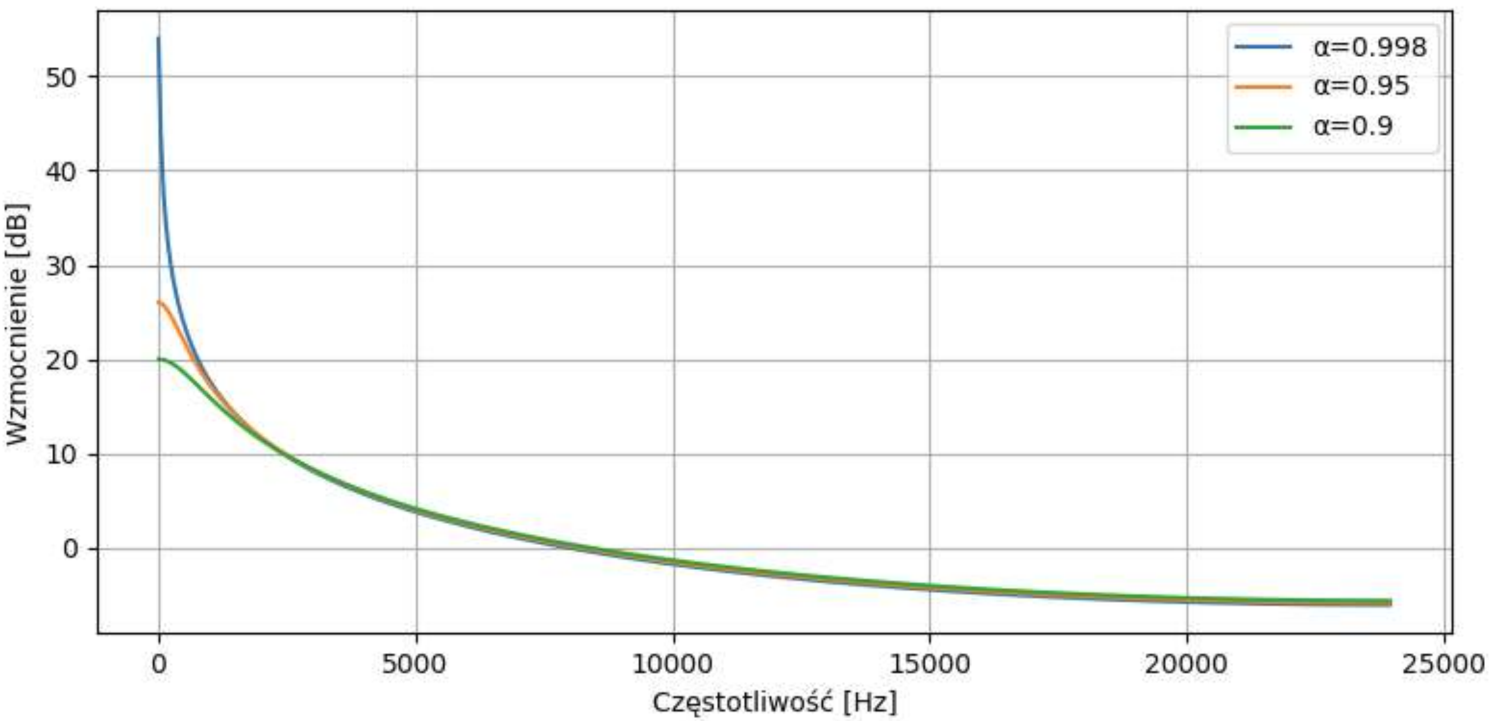
$$y[n] = x[n] + \alpha\, y[n-1]$$

- Transmitancja filtru:

$$H(z) = \frac{1}{1-\alpha z^{-1}}$$

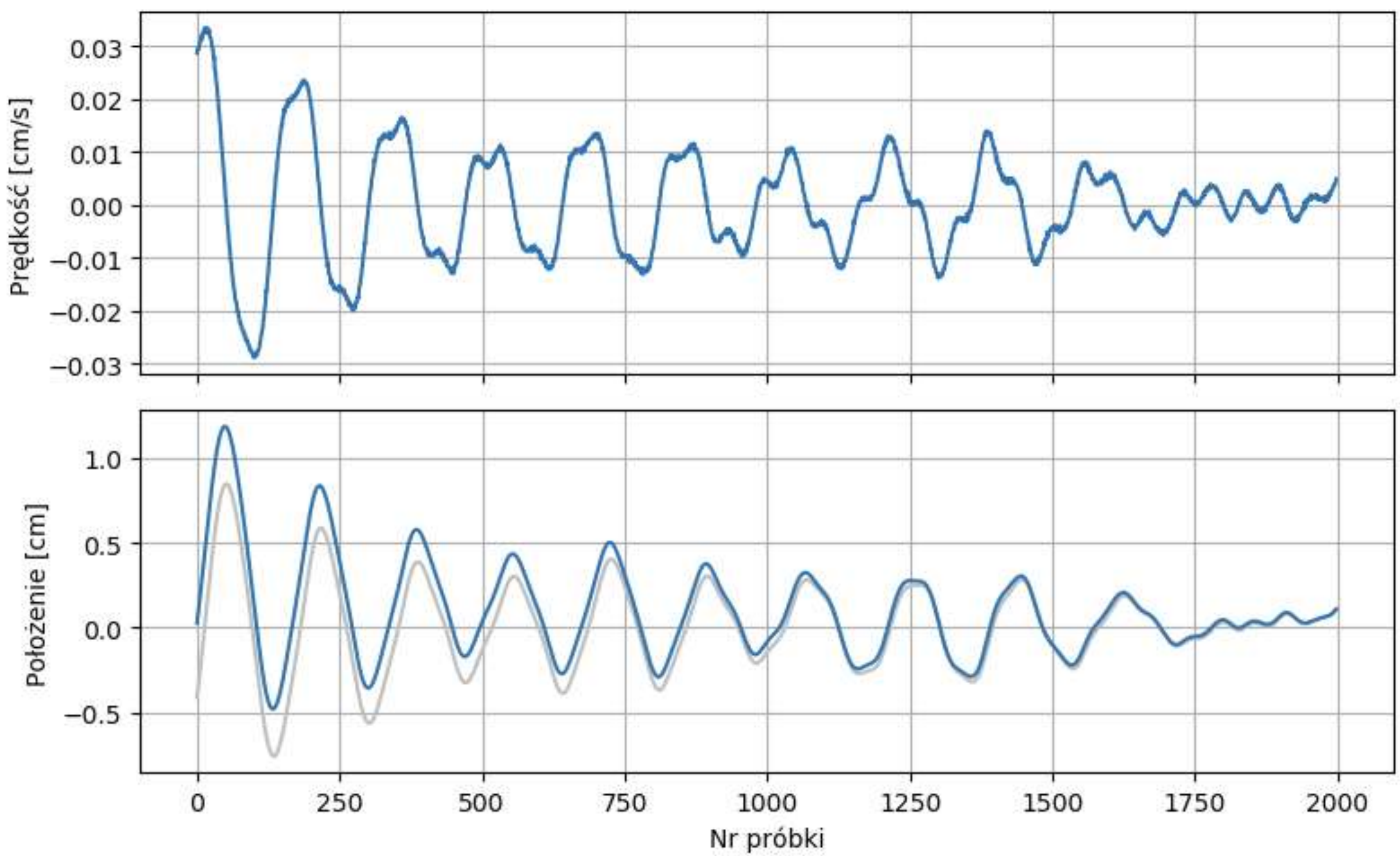
- Współczynnik musi być |α|<1 aby filtr był stabilny.
- Zwykle α jest bliskie 1, np. 0,998.
- Jest to układ *leaky integrator* („z przeciekiem”).

Charakterystyki układu całkującego dla różnych α - układ działa w przybliżeniu jak filtr dolnoprzepustowy.



Zastosowanie do poprzedniego przykładu: całkujemy sygnał prędkości, dostajemy z powrotem położenie.

Błędy na początku – filtr nie zna wcześniejszych próbek.



- Składowa stała (*direct component*, DC) jest niepożądaną składową sygnału.
- Jest ona równa wartości średniej sygnału.
- Jeżeli mamy cały sygnał do dyspozycji, wystarczy obliczyć średnią i odjąć ją od próbek.
- W widmie sygnału, składowa stała znajduje się na częstotliwości 0 (zero).
- Filtr składowej stałej (*DC blocker*) powinien więc być filtrem górnoprzepustowym, o częstotliwości granicznej bliskiej zeru i o wąskim paśmie przejściowym.

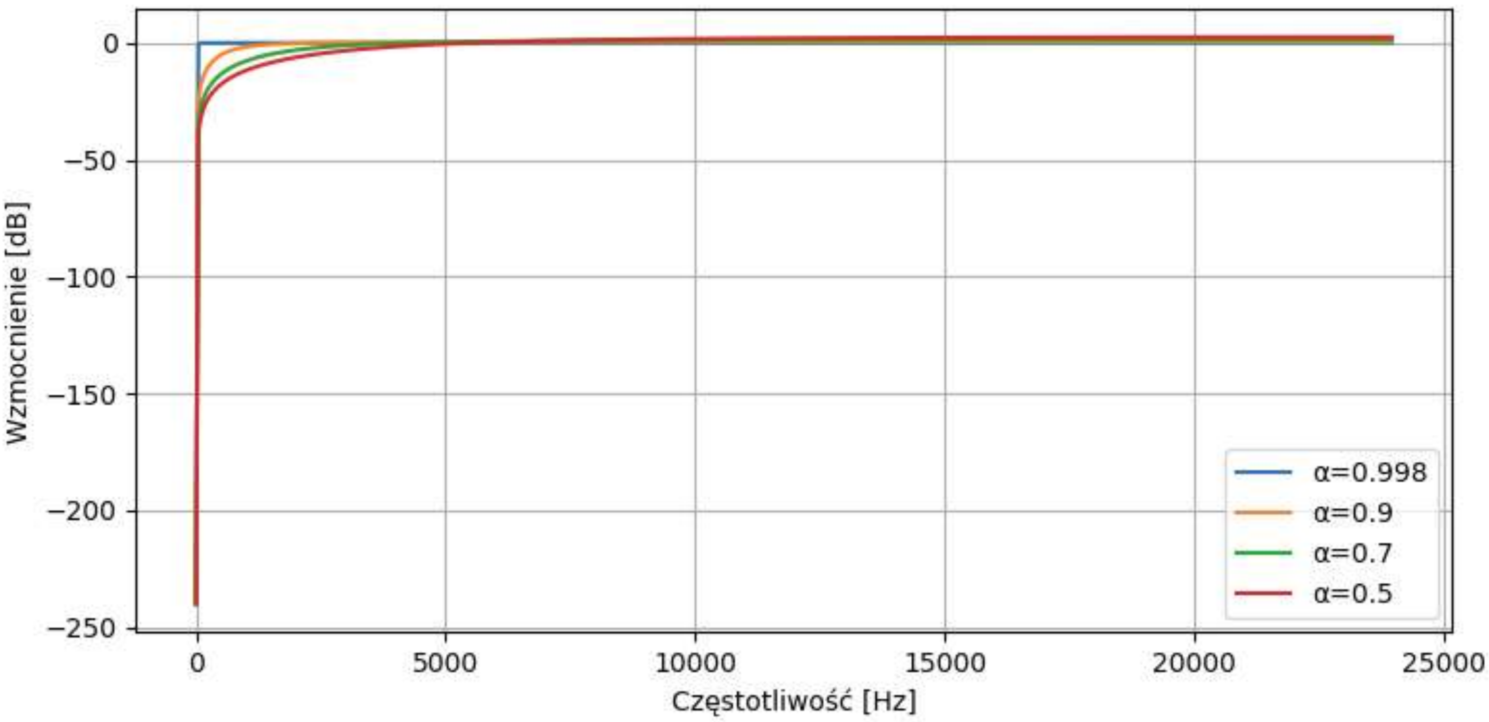
- Najprostszy, rekursywny filtr składowej stałej jest połączeniem filtru różniczkującego i całkującego:

$$y[n] = x[n] - x[n-1] + \alpha\, y[n-1]$$

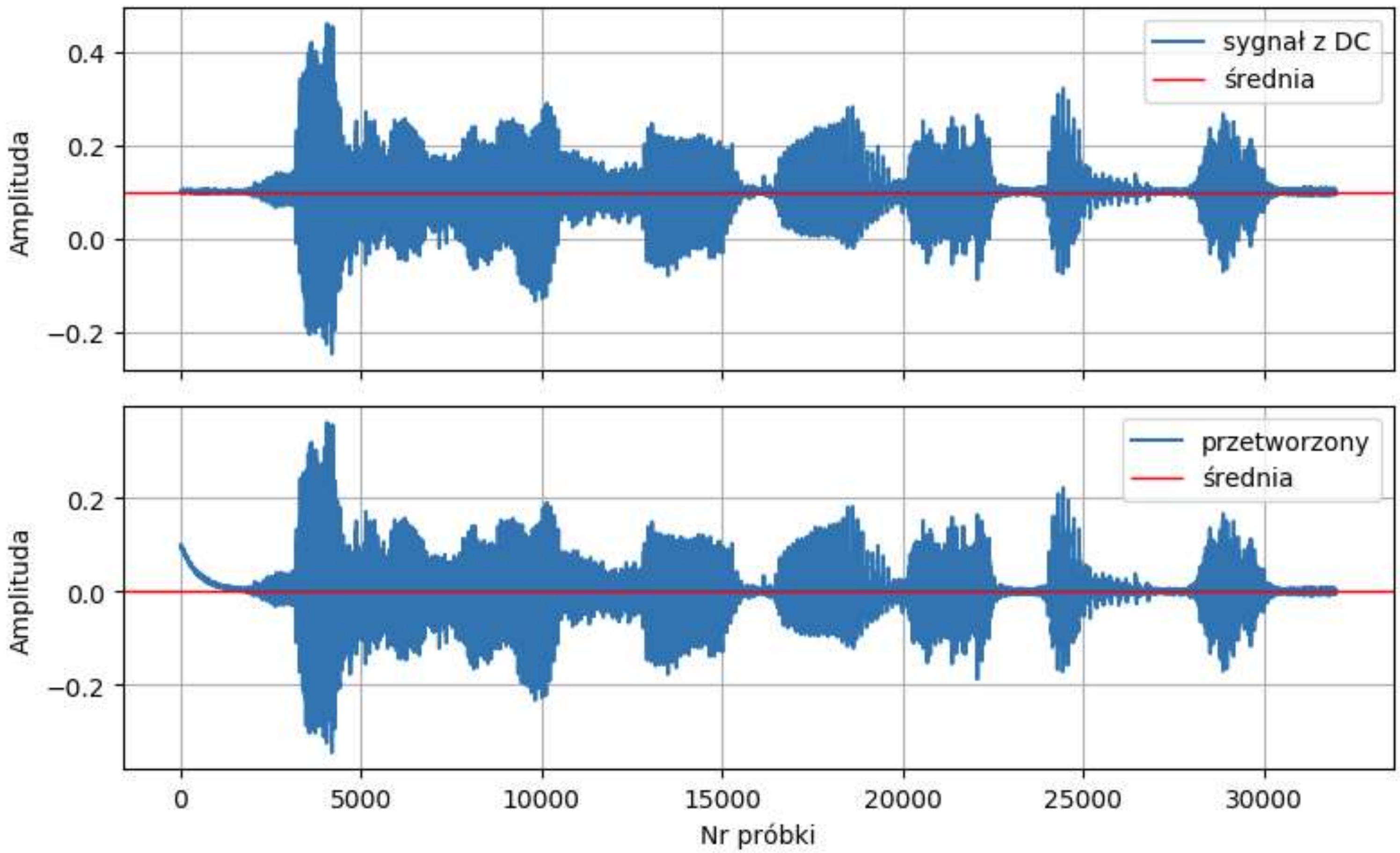
$$H(z) = \frac{1 - z^{-1}}{1 - \alpha z^{-1}}$$

- Również tutaj musi być spełniony warunek |α|<1 aby filtr był stabilny.

Charakterystyki układu usuwania DC dla różnych α - wartości bliższe 1 dają skuteczniejsze działanie.



Przykład działania: sygnał ze sztucznie dodaną składową stałą (0,1) i sygnał po filtracji (α=0.998)

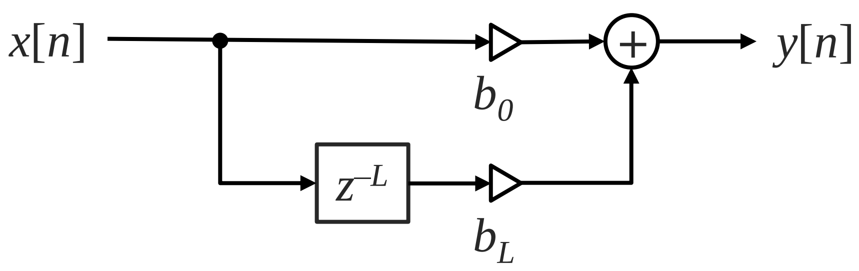


- Układ opóźniający próbki, tzw. linia opóźniająca (*delay line*), przechowuje próbki i wypuszcza je po *L* okresach próbkowania:

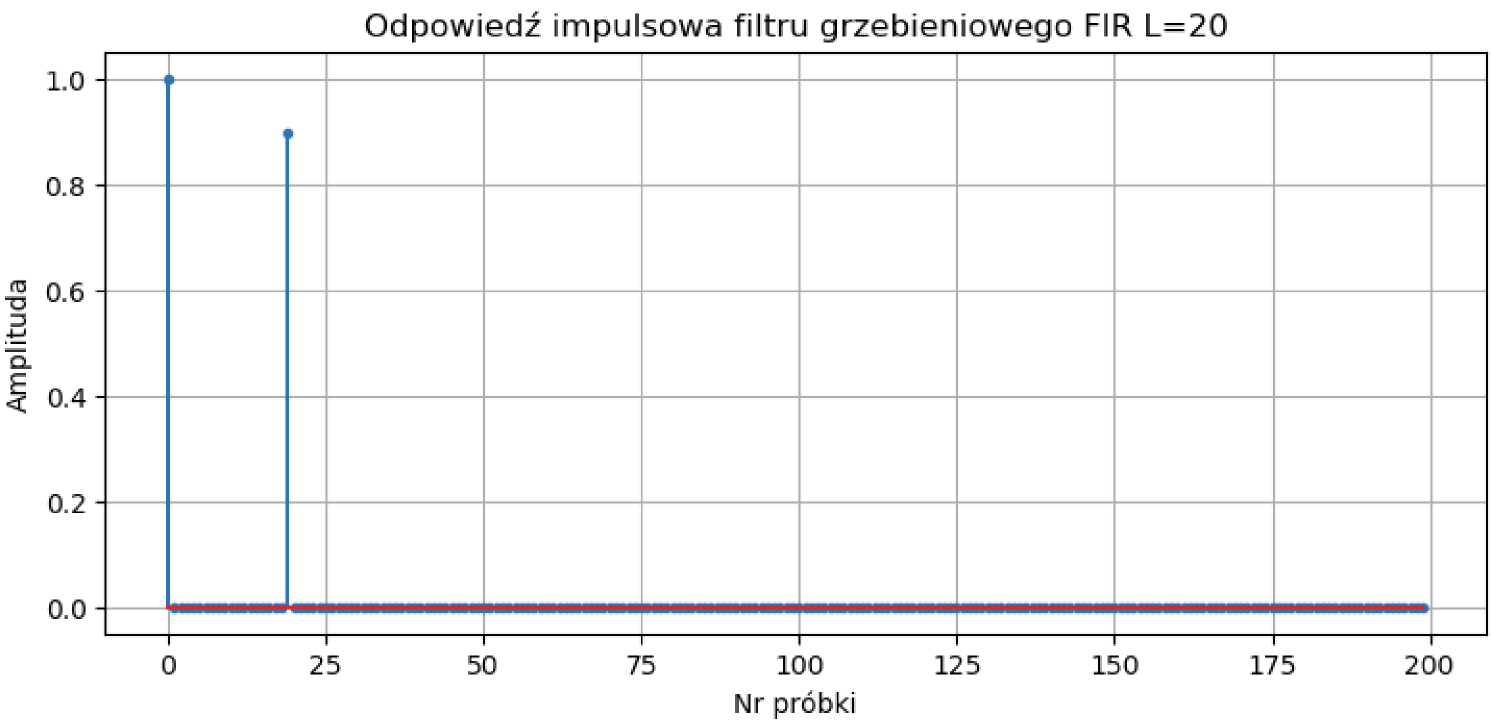
$$y[n] = x[n-L]$$

- Sama linia opóźniająca nie jest interesująca. Ale możemy do sygnału dodać jego opóźnioną i stłumioną kopię:

$$y[n] = b_0 x[n] + b_L x[n-L]$$



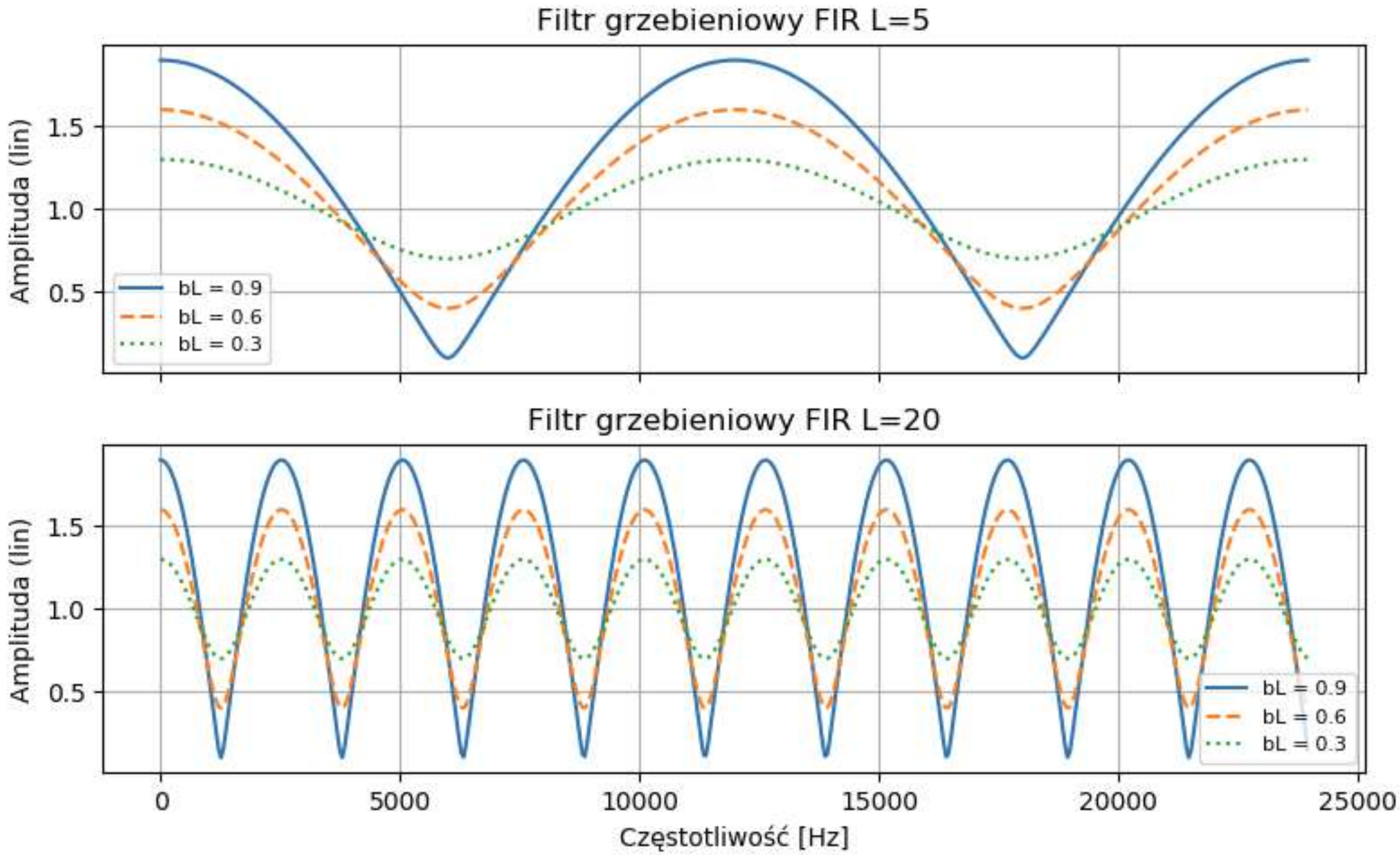
Odpowiedź impulsowa filtru:



Filtr wytwarza echo sygnału, opóźnione o *L* próbek i zmniejszone względem oryginału.

Można np. dodać echo do dźwięku.

Charakterystyki amplitudowe filtru (b_0 = 1):

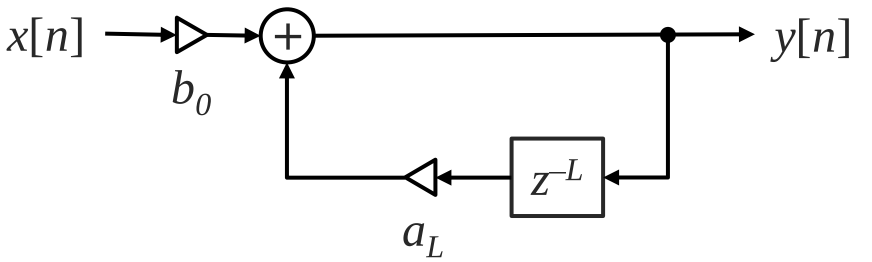


- Chcemy jedynie dodać opóźnione echo sygnału, nie chcemy zmieniać widma amplitudowego.
- Użyty filtr bardzo silnie modyfikuje to widmo.
- Ze względu na kształt widma, nazywa się go filtrem grzebieniowym (*comb filter*).
- Stosując opóźnienie o L próbek dostajemy widmo, w którym jest L „zębów grzebienia” w zakresie od 0 do f_S.
- Amplitudy maksimów zależą od współczynnika b_L.

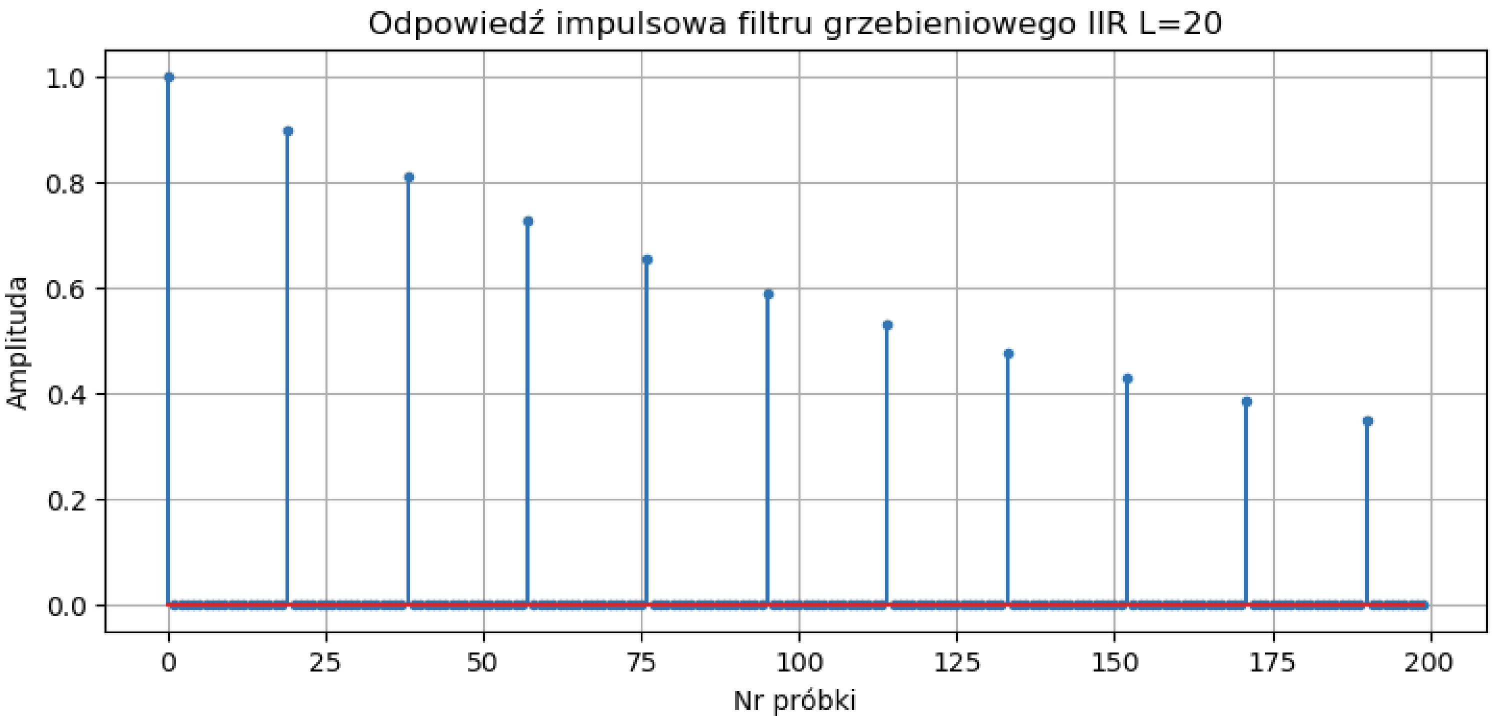
- Możemy ten sam układ zrealizować jako filtr rekursywny: do sygnału dodajemy opóźnione o *L* próbek wyjście filtru:

$$y[n] = b_0 x[n] + a_L y[n-L]$$

- Aby filtr był stabilny: $|a_L| < 1$.



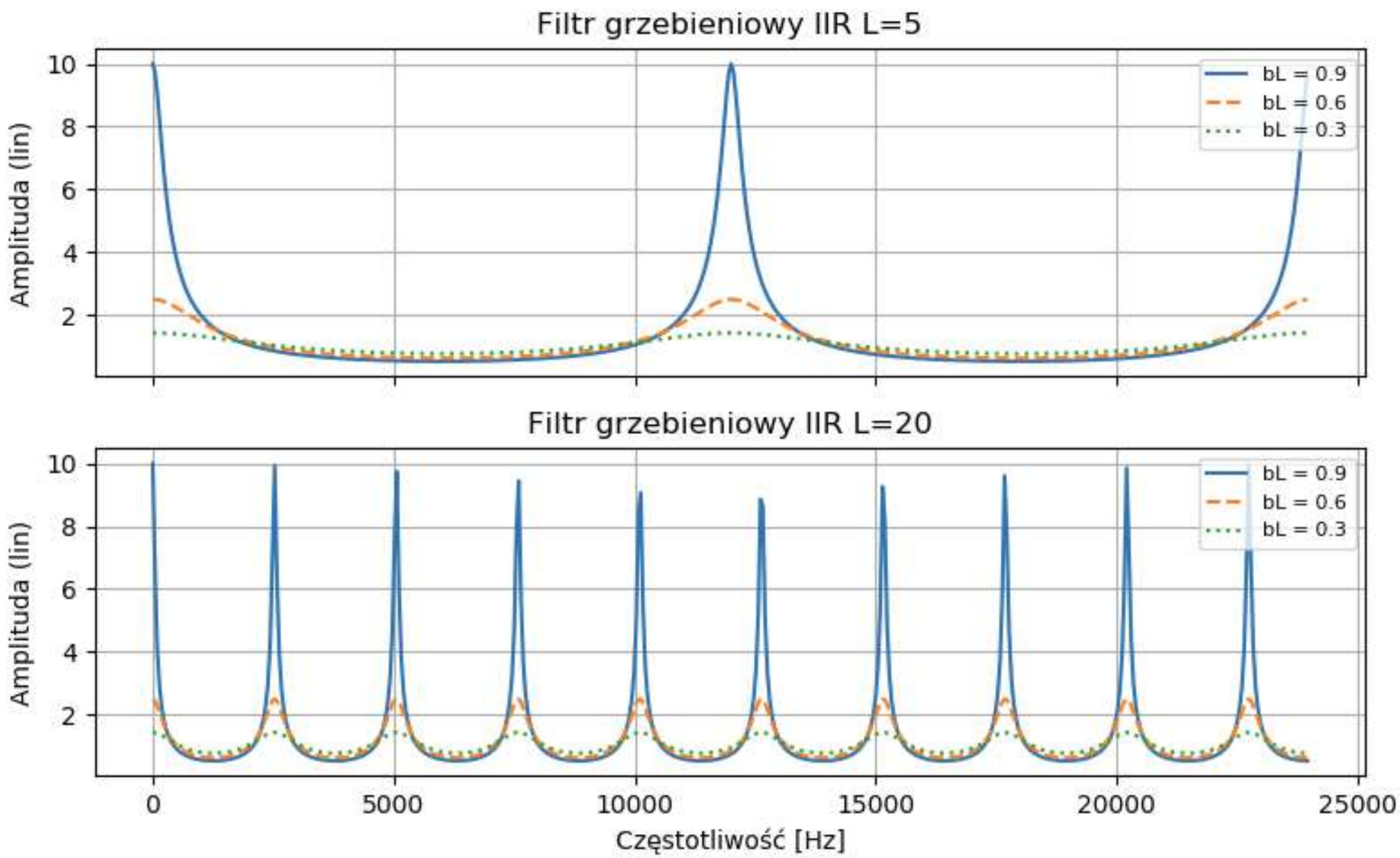
Odpowiedź impulsowa filtru rekursywnego:



Filtr wytwarza wielokrotne echo: tłumione kopie sygnału pojawiają się co L próbek.

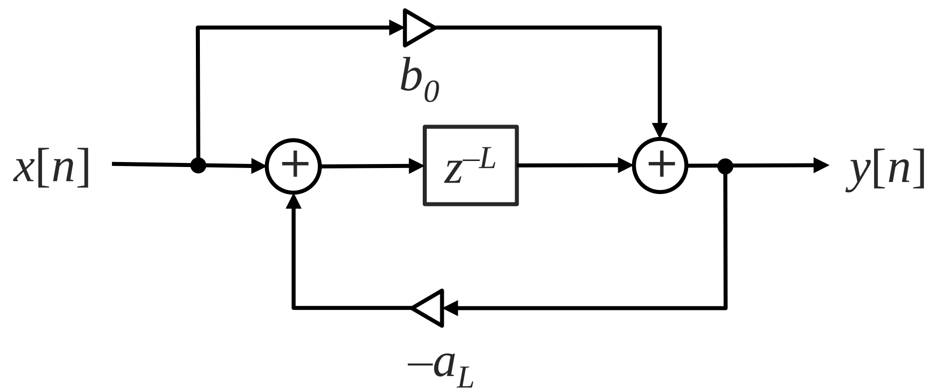
Również nadaje się jako efekt dźwiękowy.

Charakterystyki amplitudowe filtru ($b_0 = 1$): również jest to filtr grzebieniowy, który ma „odwrotnie zęby” niż filtr FIR.



- Oba filtry grzebieniowe (FIR i IIR) zniekształcają widmo (efekt grzebienia).
- Chcemy mieć układ, który generuje echo (opóźnia sygnał), ale nie modyfikuje widma amplitudowego.
- Kształt widma filtrów FIR i IIR jest w przybliżeniu przeciwstawny.
- Idea: a może połączyć oba typy filtrów w jeden? Czy to zadziała tak jak chcemy?

Układ z dwoma pętlami:



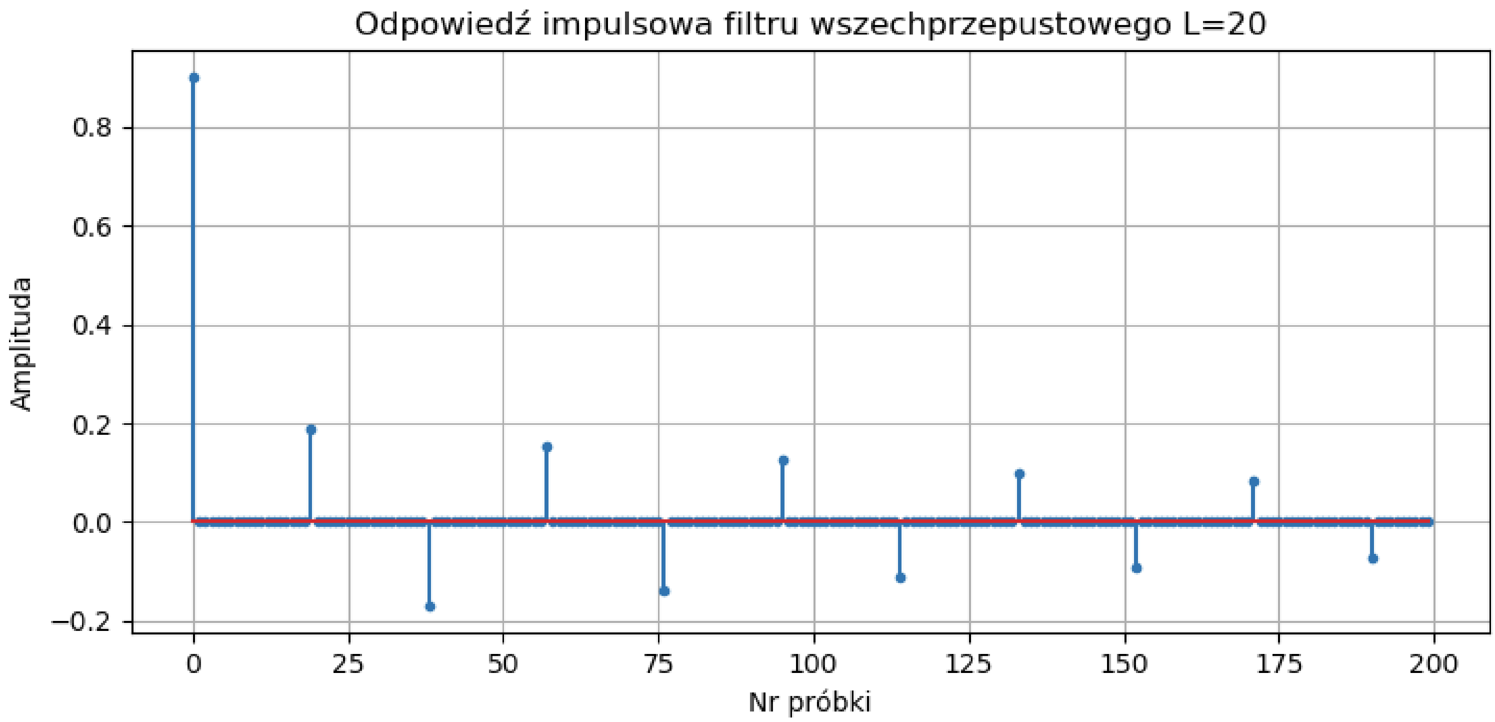
Równanie filtru:

$$y[n] = b_0 x[n] + x[n-L] - a_L y[n-L]$$

Transmitancja:

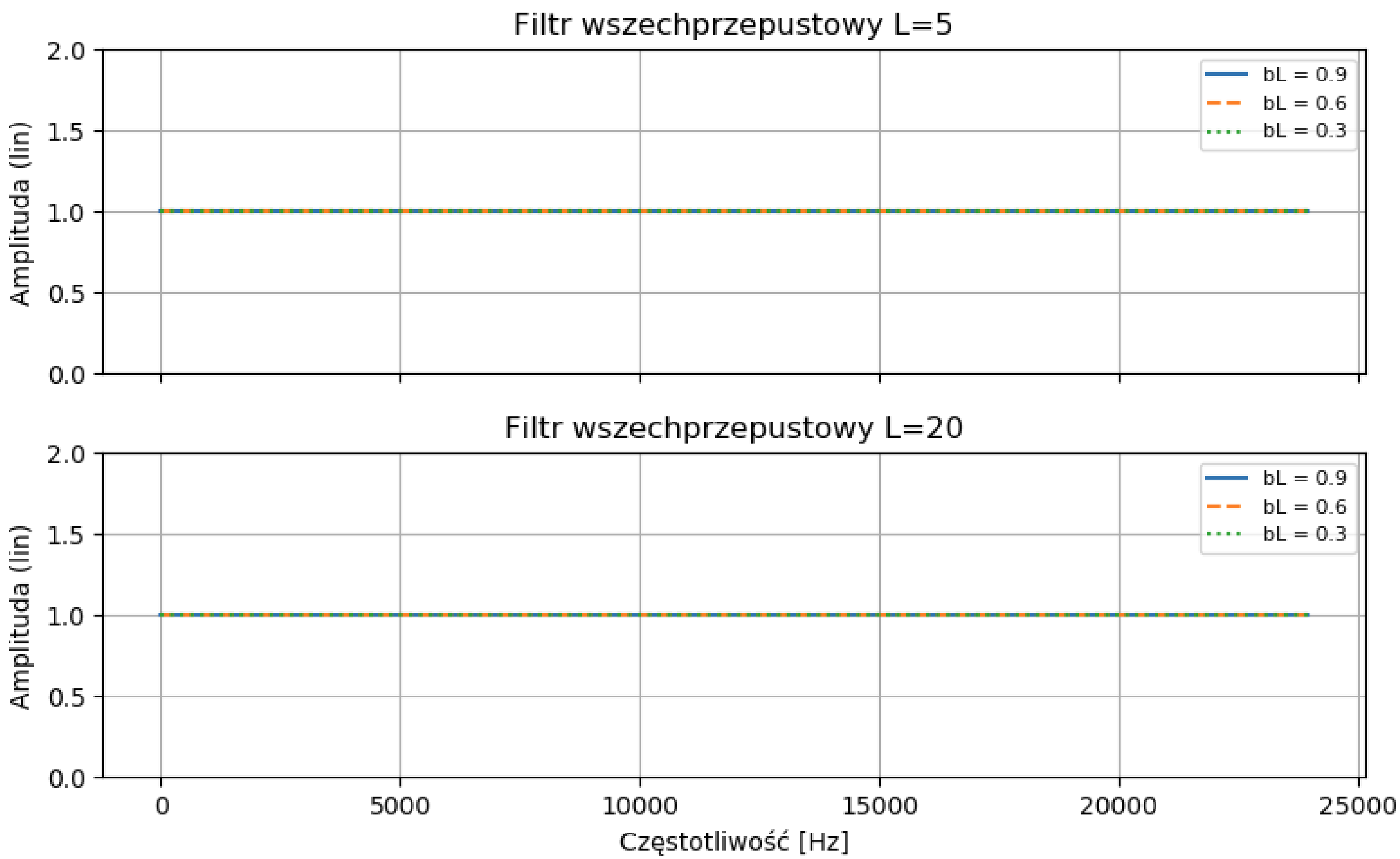
$$H(z) = \frac{b_0 + z^{-L}}{1 + a_L z^{-L}}$$

Odpowiedź impulsowa:



Filtr nadal wytwarza wielokrotne echo, tak jak chcemy.

Charakterystyki częstotliwościowe:



Widmo ampl. = 1 w całym zakresie częstotliwości!

- Omawiany układ przenosi wszystkie częstotliwości bez zmiany amplitudy.
- Z tego powodu jest nazywany filtrem wszechprzepustowym (*allpass filter*).
- Modyfikuje tylko charakterystykę fazową, wytwarza wielokrotne echo.
- Układ działa, pod warunkiem że $b_0 = a_L$. Wtedy wpływy obu pętli znoszą się nawzajem.
- Filtr wszechprzepustowy jest wykorzystywany m.in. do efektów brzmieniowych (*delay*, *reverb*, *phaser*) oraz w korektorach barwy dźwięku.

- W typowych filtrach cyfrowych trudno jest modyfikować kształt charakterystyki, np. stopień tłumienia – zależy od wszystkich współczynników.
- W filtrach analogowych jest to proste (zmienna rezystancja – potencjometr).
- Filtry parametryczne (*parametric filters*) pozwalają modyfikować charakterystykę (częstotliwość graniczna, wzmocnienie/tłumienie) za pomocą pojedynczych parametrów.
- Cyfrowe filtry parametryczne są obliczane na podstawie analogowych filtrów. Są to filtry rekursywne (IIR), więc mają wszystkie ich wady.

Filtry półkowe (*shelving filters*)

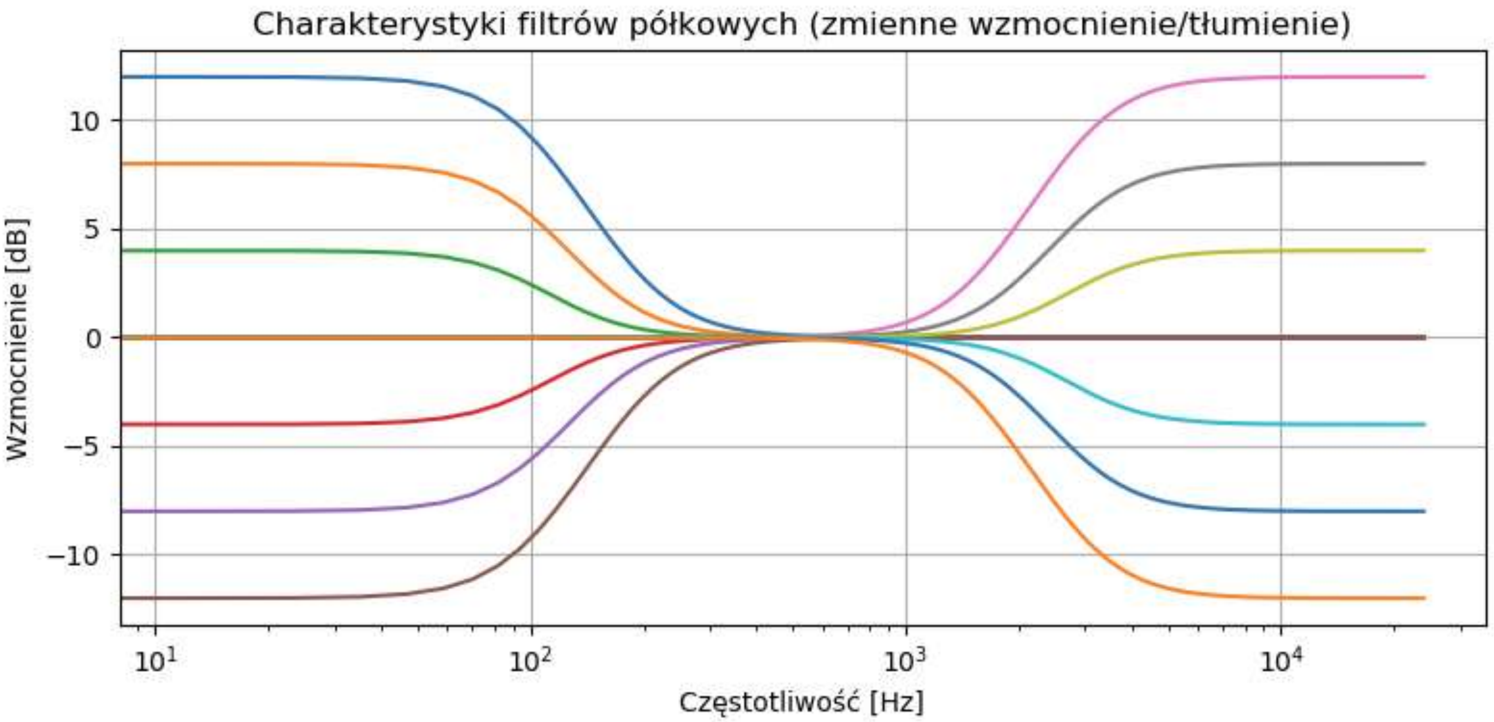
- Filtry, które:
  - przepuszczają jeden zakres cz. bez zmian,
  - wprowadzają wzmocnienie (*boost*) lub tłumienie (*cut*) w pozostałej części pasma.
- Filtry modyfikują albo niskie, albo wysokie częstotliwości.
- Parametry, które można zmieniać:
  - częstotliwość graniczna,
  - stopień wzmocnienia ($g > 1$) lub tłumienia ($g < 1$) w modyfikowanym paśmie.

- Filtr półkowy modyfikujący dolne (PD) lub górne (PG) częstotliwości jest równoległym połączeniem (czyli sumą) filtru wszechprzepustowego (WP) z filtrem dolno- lub górnoprzepustowym:

$$PD = WP + (g - 1)DP \qquad PG = WP + (g - 1)GP$$

- W ten sposób uzyskuje się wzmocnienie. Aby stłumić pasmo, odwraca się transmitancję filtru.
- Taki analogowy prototyp przekształca się w formę cyfrową, np. za pomocą przekształcenia dwuliniowego.

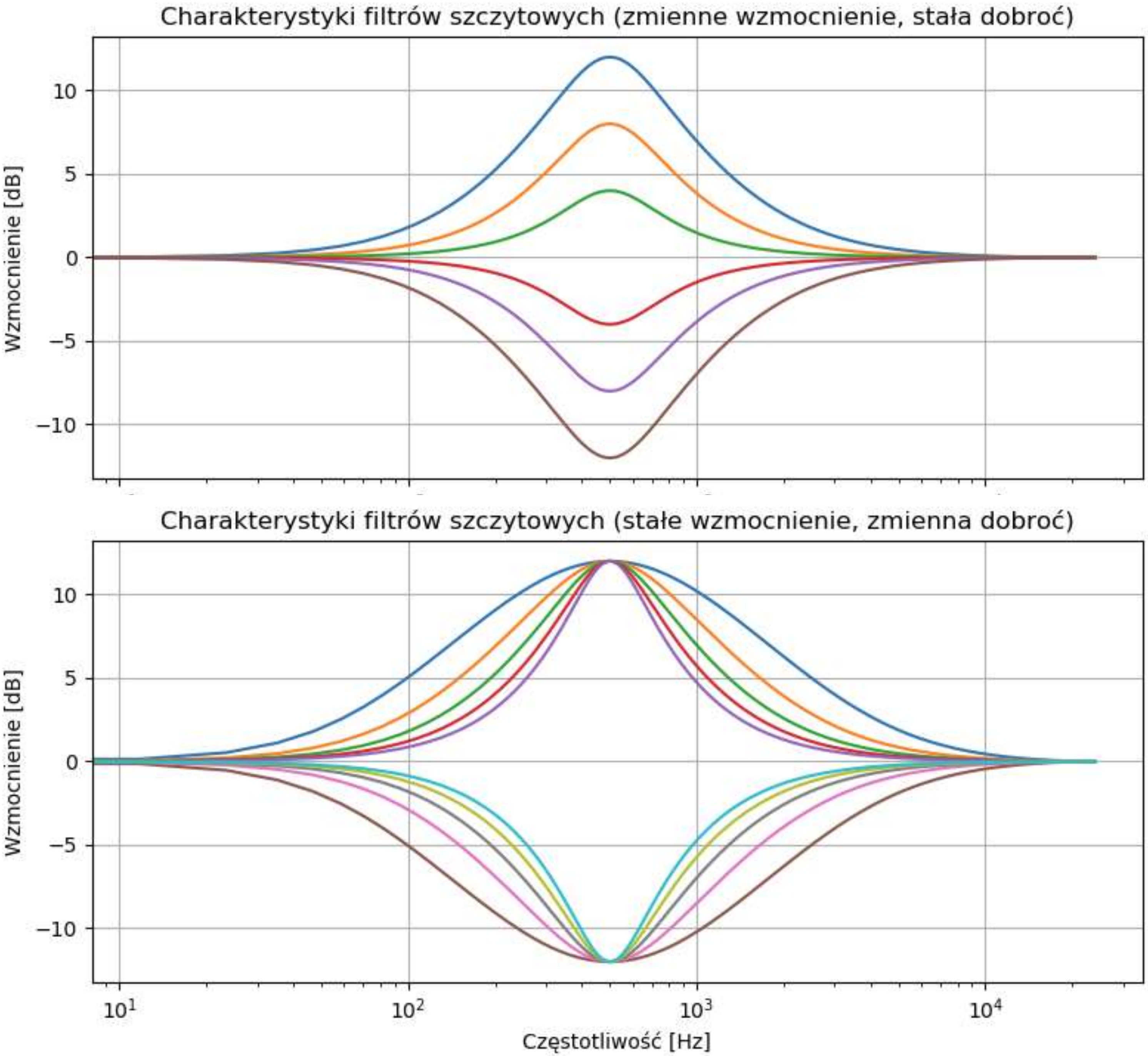
Przykład zastosowania filtrów półkowych: regulator niskich i wysokich „tonów” (korekcja barwy dźwięku)



Filtr szczytowy (*peak filter*)

- Parametryczny filtr pasmowo-przepustowy.
- Reguluje wzmocnienie dla środkowych częstotl.
- Suma filtru wszechprzepustowego i PP.
- Parametry: częstotliwość środkowa, wzmocnienie oraz dobroć (Q) – stosunek cz. środkowej f_c do szerokości pasma *B* – różnica częstotliwości, dla których wzmocnienie jest o 3 dB mniejsze niż w f_c.

$$Q = \frac{f_c}{B} = \frac{f_c}{f_g - f_d}$$



Charakterystyki filtrów szczytowych (zmienne wzmocnienie, stała dobroć)
Wzmocnienie [dB]
10
5
0
−5
−10
Charakterystyki filtrów szczytowych (stałe wzmocnienie, zmienna dobroć)
Wzmocnienie [dB]
10
5
0
−5
−10
10^1
10^2
10^3
10^4
Częstotliwość [Hz]

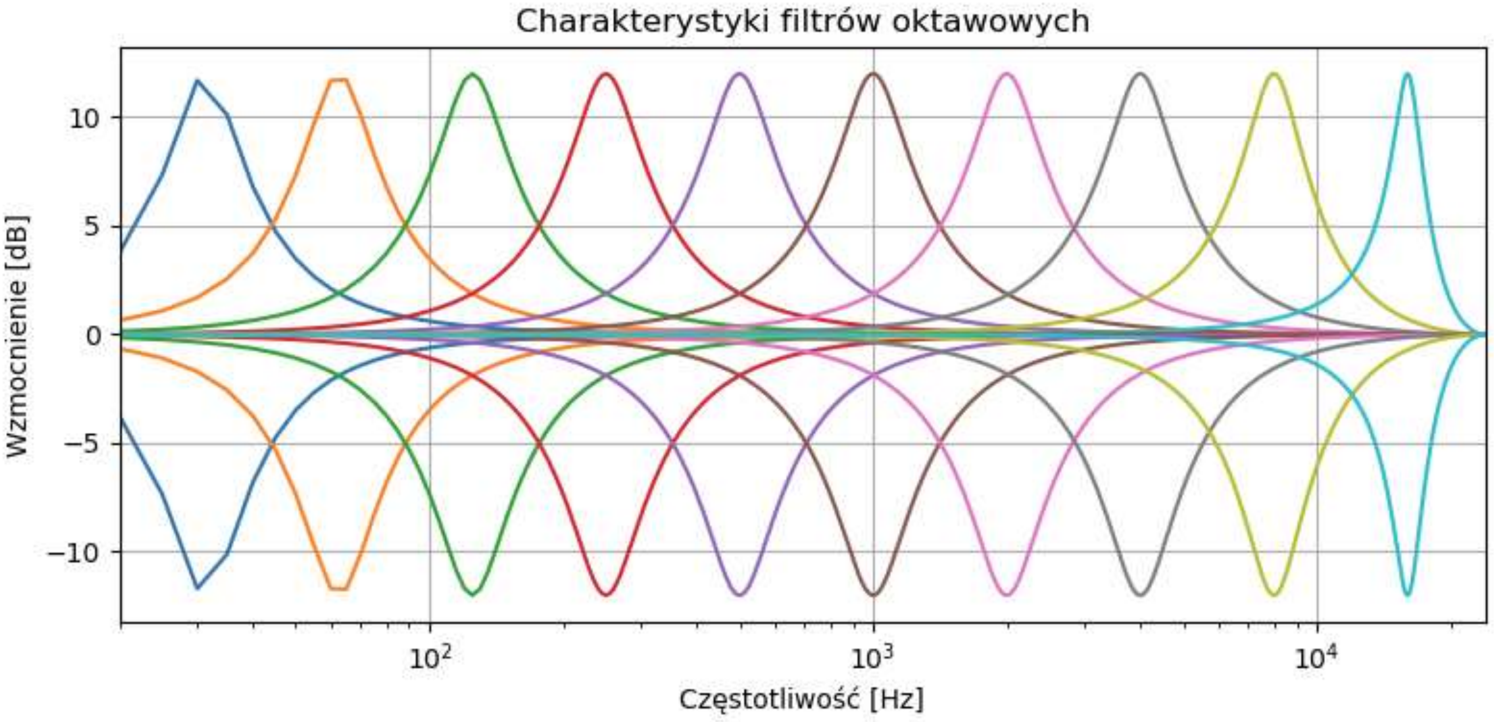
Bank filtrów (*filter bank*)

- Zbiór filtrów (np. szczytowych) pokrywających cały zakres częstotliwości.
- Suma filtrów o jednakowym wzmocnieniu powinna dać płaską charakterystykę widmową.
- Przykład banku: filtry oktawowe (*octave filters*). Stosunek częstotliwości środkowych sąsiednich filtrów jest równy oktawie, czyli 2.
- Są to filtry o stałej dobroci (*constant Q*)

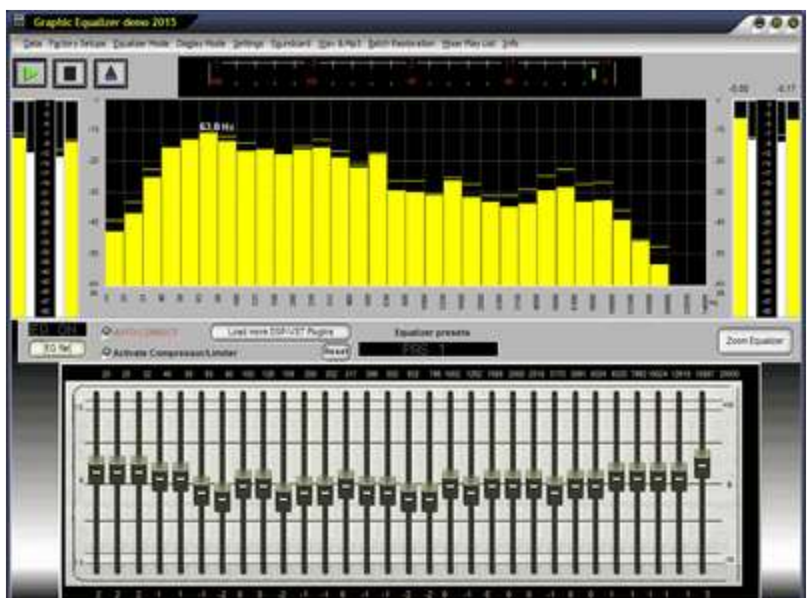
$$\frac{f_{c(n+1)}}{f_{c(n)}} = \frac{f_{g(n)}}{f_{d(n)}} = 2$$

$$Q = \frac{\sqrt{2}}{\sqrt{2}-1} \approx 3{,}414214$$

Charakterystyki banku filtrów oktawowych (wzmocnienie ±12 dB)



Przykład praktycznego zastosowania:
korektor graficzny dźwięku (*graphic equalizer*)

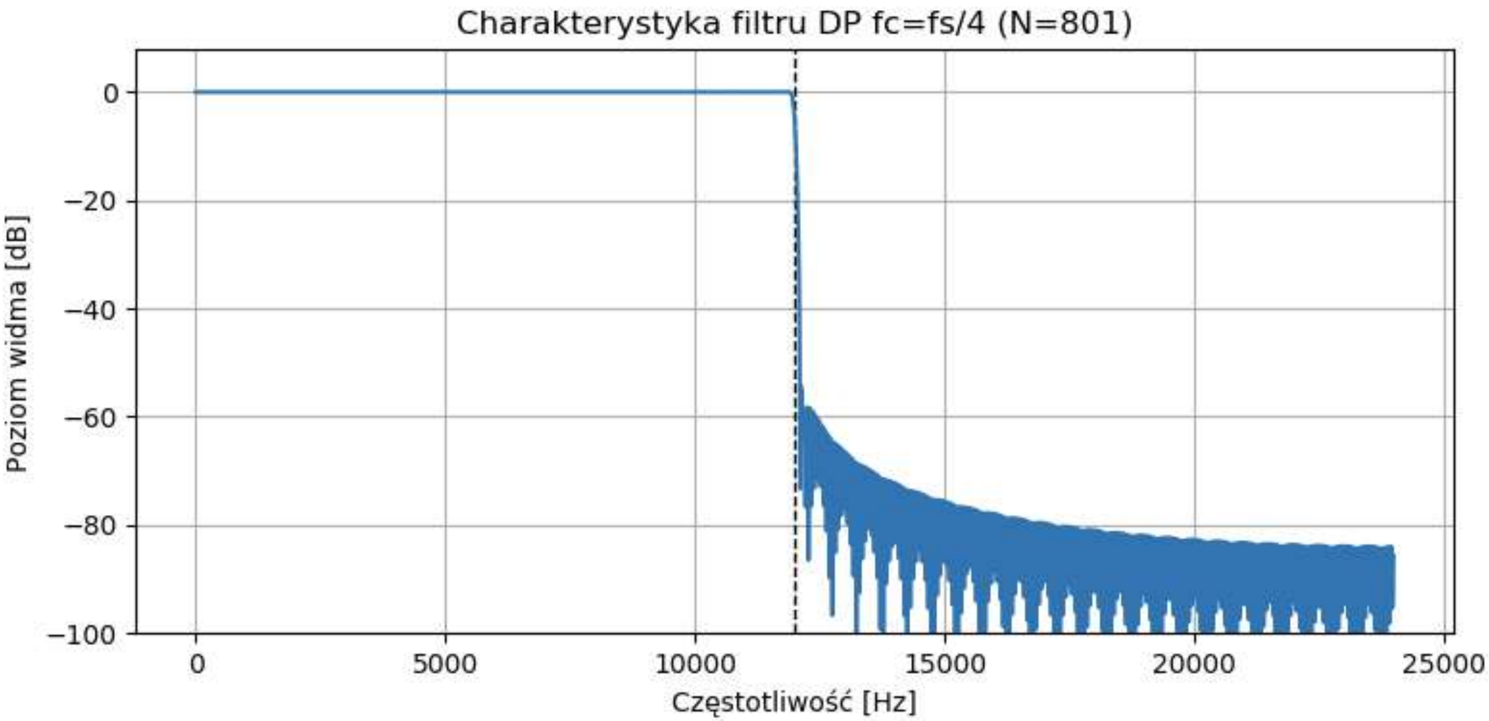


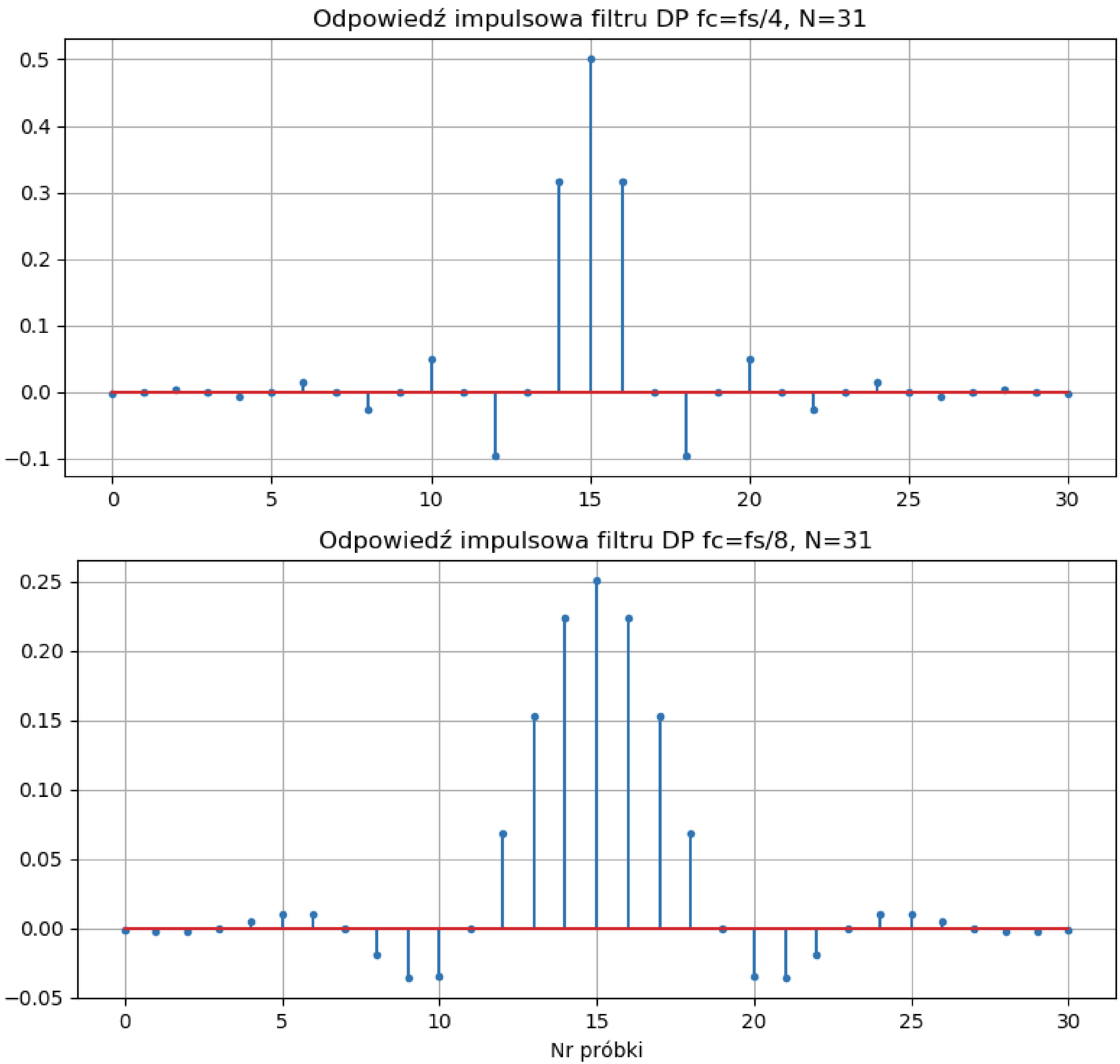
- Specjalną grupę filtrów stanowią filtry dolnoprzepustowe, których częstotliwość graniczna jest całkowitym ułamkiem cz. Nyquista:

$$f_c = \frac{f_s}{2m}$$

- Dla *m* = 2 mamy filtr połówkowy (*half-band filter*) - pasmo przepustowe zajmuje połowę całkowitego pasma ($f_c = f_s / 4$).
- Filtry FIR tego typu mają ważną cechę: co *m*-ta próbka odpowiedzi impulsowej jest zerowa!
- Można więc zredukować liczbę wykonywanych operacji mnożenia (f. połówkowy: dwukrotnie).

Charakterystyka widmowa filtru połówkowego FIR:




Odpowiedź impulsowa filtru DP fc=fs/4, N=31
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
0
5
10
15
20
25
30
Odpowiedź impulsowa filtru DP fc=fs/8, N=31
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
0
5
10
15
20
25
30
Nr próbki

Zmniejszenie częstotliwości próbkowania sygnału nazywa się decymacją (*decimation*).

Przykład: decymacja z 48 kHz do 16 kHz: ze stopniem 3.

Będziemy rozpatrywać decymację ze stopniem M, który jest liczbą całkowitą, np. $M = 3$.

Decymacja z $fs1$ do $fs2 = fs1 / M$:

- filtracja dolnoprzepustowa – filtr decymacyjny o częstotliwości granicznej $f_c = fs2 / 2$; konieczny, aby zapobiec aliasingowi widma,
- pobranie z wyniku filtracji co M-tej próbki, odrzucając pozostałe próbki.

Zwiększenie częstotliwości próbkowania sygnału nazywa się interpolacją (*interpolation*).

Np. interpolacja z 48 kHz do 96 kHz: ze stopniem 2.

Będziemy rozpatrywać interpolację ze stopniem L, który jest liczbą całkowitą, np. $L = 2$.

Interpolacja z $fs1$ do $fs2 = fs1 \cdot L$:

- wstawienie L-1 zer między każdą parę próbek oryginalnego sygnału,
- filtracja dolnoprzepustowa – filtr interpolacyjny o częstotliwości granicznej $f_c = fs1 / 2$, usuwa „nadmiarowe” kopie widma.

Rozpatrzmy konwersję częstotliwości próbkowania z 48 kHz do 32 kHz.

48 / 32 = 3 / 2 = M / L. A więc:

- wstawiamy L-1 = 1 zero między każdą parę próbek (interpolacja), mamy $fs = 2 \cdot 48 = 96$ kHz,
- wykonujemy filtrację dolnoprzepustową filtrem o częstotliwości $f_c = 32/2 = 16$ kHz,
- bierzemy co trzecią próbkę z wyniku (decymacja).

Zauważmy, że wystarczy jeden filtr. Nie ma potrzeby stosowania dwóch filtrów ($f_c = 24$ kHz i $f_c = 16$ kHz).

Praktyczny przykład: mamy nagranie dźwiękowe spróbkowane z 44,1 kHz. Potrzebujemy 48 kHz.

- Szukamy najmniejszego stosunku liczb całkowitych równego stosunkowi częstotliwości:

$$\frac{f_{s2}}{f_{s1}} = \frac{48}{44{,}1} = \frac{160}{147}$$

- Wstawiamy 159 zer między każdą parę próbek.
- Przepuszczamy sygnał przez filtr DP o f_c = 22,05 kHz.
- Bierzemy co 147 próbkę z wyniku.

Jest to bardzo mało efektywne podejście, w praktyce stosuje się np. przepróbkowanie w dziedzinie widma.

Wróćmy do problemu interpolacji ze stopniem 2.

- Wstawiamy zero między każdą parę próbek, po czym filtrujemy.
- Zauważmy jednak, że zawsze połowa współczynników filtru (na zmianę: „parzysta” i „nieparzysta”) jest mnożona przez zera:

h0	h1	h2	h3	h4	h5	h6	h7
x(0)	0	x(-1)	0	x(-2)	0	x(-3)	0

h0	h1	h2	h3	h4	h5	h6	h7
0	x(0)	0	x(-1)	0	x(-2)	0	x(-3)

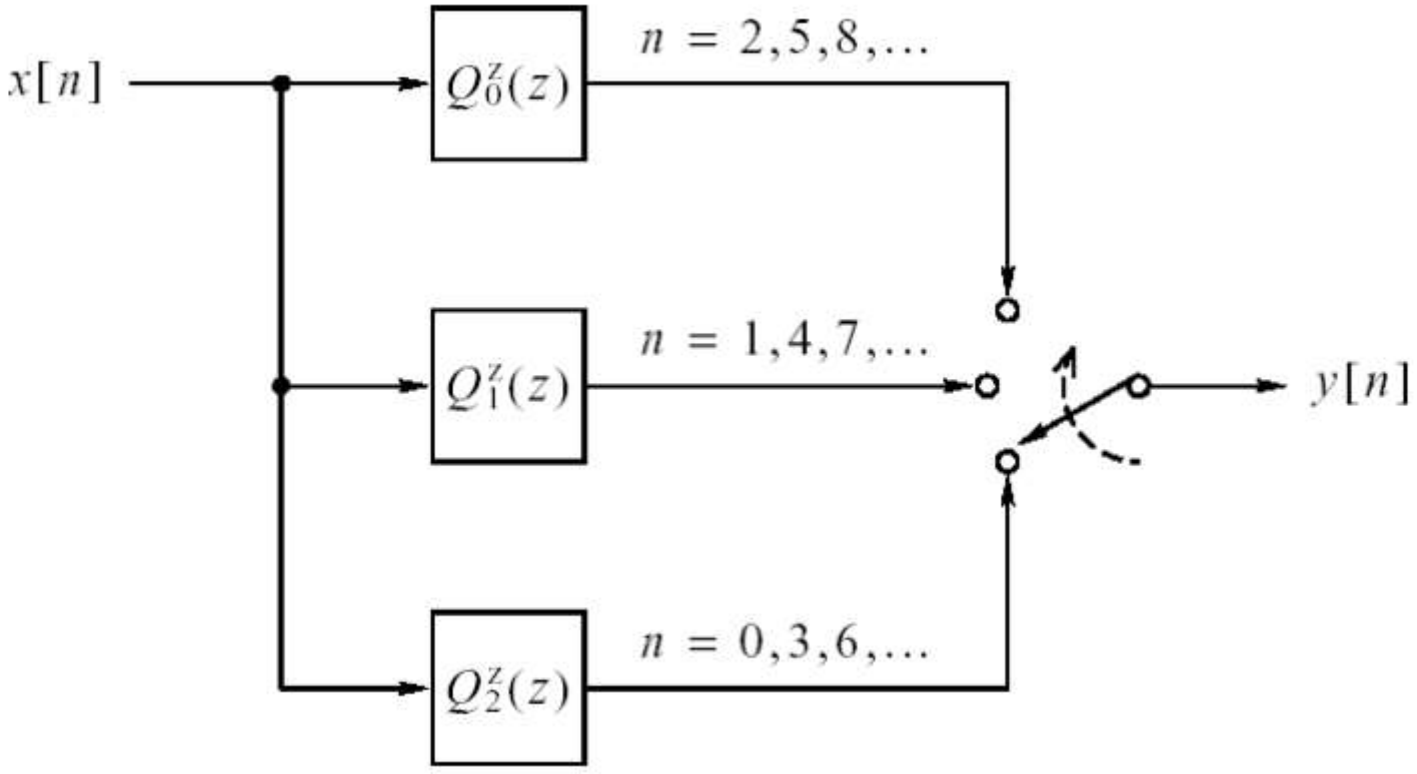
Optymalizacja interpolacji ze stopniem 2:

- nie wstawiamy żadnych zer do sygnału,
- dzielimy filtr na dwie części: „parzystą” i „nieparzystą”,
- każdą próbkę sygnału przetwarzamy najpierw przez filtr „parzysty”, a potem „nieparzysty”,
- mamy dwie próbki sygnału po filtracji i interpolacji.

Taką samą metodę możemy zastosować do interpolacji z dowolnym stopniem L – dzielimy filtr na więcej części.

Opisana metoda jest strukturą filtru polifazowego (*polyphase filter*).

Struktura dla interpolacji ze stopniem 3:

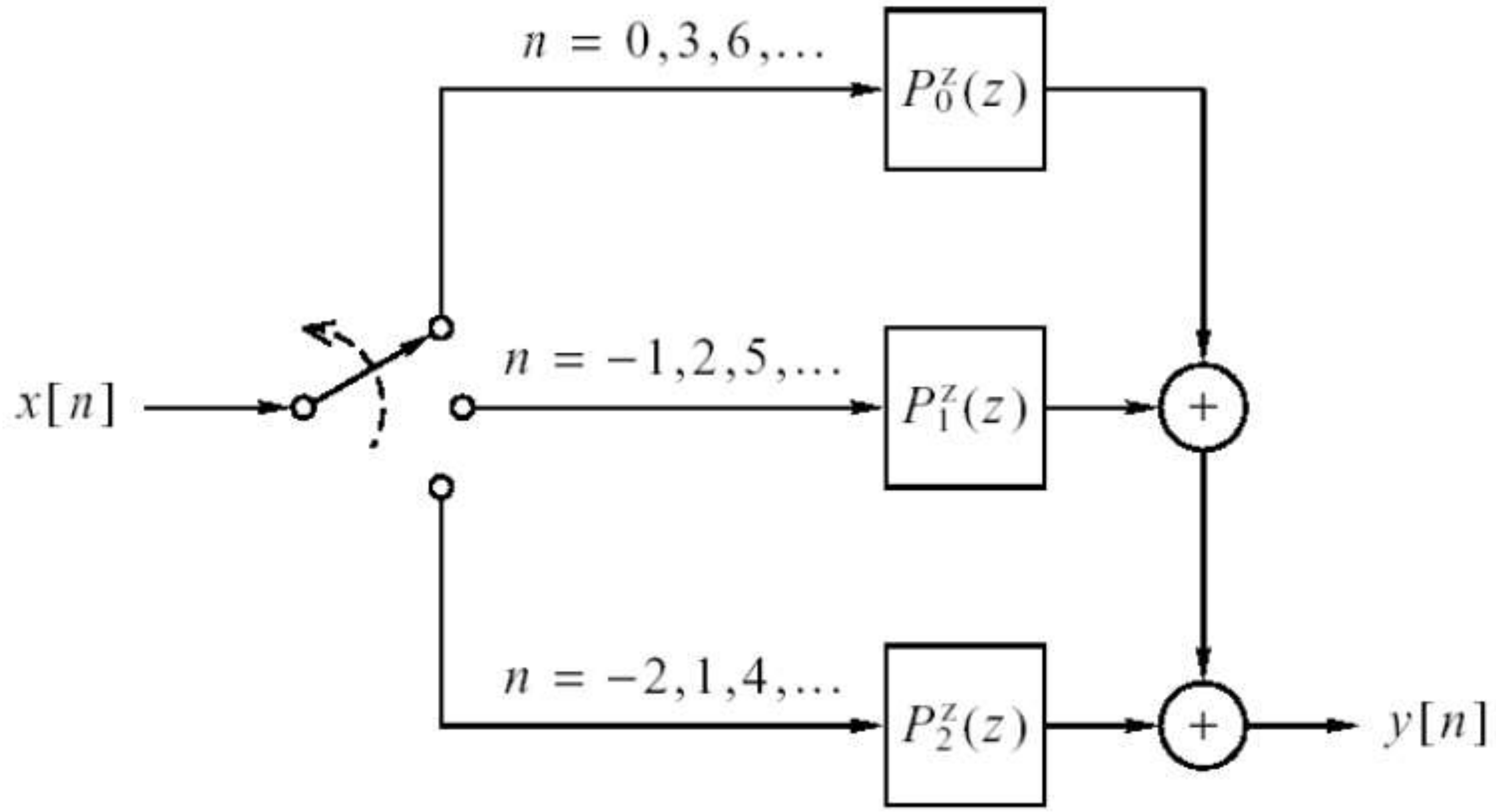


Podobnie możemy postąpić dla decymacji
ze stopniem 2:

- dzielimy filtr na dwie części: „parzystą” i „nieparzystą”,
- bierzemy parę próbek wejściowych, pierwszą przetwarzamy przez filtr „parzysty”, drugą przez „nieparzysty”,
- wyniki z obu filtrów sumujemy – mamy jedną próbkę wyjściową (po filtracji i decymacji) z dwóch wejściowych.

Taką samą metodę możemy zastosować do decymacji
z dowolnym stopniem M.

Struktura filtru polifazowego dla decymacji ze stopniem 3:



Wykorzystanie filtru polifazowego do konwersji częstotliwości próbkowania z 48 kHz do 32 kHz:

- $M / L = 48 / 32 = 3 / 2$.
- Projektujemy filtr dolnoprzepustowy o częstotliwości $f_c = 32 / 2 = 16$ kHz.
- Dzielimy go na dwie części („parzysta” i „nieparz.”).
- Każdą próbkę wejściową przetwarzamy przez oba filtry (interpolacja) – mamy $fs = 96$ kHz.
- Bierzemy co trzecią próbkę z wyniku (decymacja) – mamy $fs = 32$ kHz, z pasmem ograniczonym do 16 kHz.